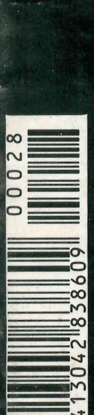
USER ATARI

AÑO III - Nº 28 - 1.991 / 375 ptas. (inc. IVA)

LA 1ª PUBLICACION PARA USUARIOS DE ATARI

- **GRAFICOS**
  LA PERSPECTIVA CONICA
- **ESCUCHANDO TU STE**
  COMO HACER SONAR AL LMC
- **PSION ORGANISER II**
  EL MINI-SUPER PERIFERICO
- **LENGUAJE C Y CODIGO MAQUINA**
  TUS CURSOS DE PROGRAMACION
- **JUEGOS**
  LAS ULTIMAS NOVEDADES



ATARI USER es una publicación de:



ATARI USER 28
JULIO - AGOSTO

DIRECTOR
Luis García Sánchez
ASESOR EDITORIAL
Pablo Sáez de Hoyos
REDACCION
Santiago Vernes
Jorge V.S.J.
Eduardo Torres
DISEÑO Y MAQUETA
José Luis Martínez
COORDINACION
Conchi G. Otero
COLABORADORES
A. Miguel Zuñiga
Fernando Perla
Rafael Fornies
R. Lucas Rotger
PUBLICIDAD
Begoña Gómez
Tlfno. (91) 639 49 20
Fax. (91) 639 51 34
SUSCRIPCIONES
CORRESPONDENCIA
COLABORACIONES
Los Altos del Burgo
Ecija, 52.
28230 LAS ROZAS
MADRID

Editorial

Sí amigos, estamos en pleno verano, la mayoría de vosotros seguro que estáis disfrutando las vacaciones, otros no tenemos tanta suerte.

Los que estéis desocupados ya podéis poneros manos a la obra pues en este número tenéis mucho trabajo, empezando por el nuevo curso de Código Máquina y continuando con las Interrupciones y la Guía Completa del Lenguaje C. También comprendemos que es tiempo de ocio y que más de uno ya tenéis en vuestro poder los divertidos juegos que están apareciendo esta temporada, desde el Skull & Crossbones al entretenido y enloquecedor Lemmings. En breve aparecerán, junto a los comentarios de los juegos, todos los trucos y pokes posibles, para que esos pequeños problemillas (vidas, tiempo, bonus,...) que os surgen los podáis solventar y hacer así el juego más divertido (o menos desesperante, según se mire). A los devotos de los gráficos, en este número se hace una pequeña introducción a la Perspectiva Cónica. ¡Ah!, también le ponemos oídos al STE.

Por otro lado os queremos recordar que nuestra única meta es hacer una revista cada vez mejor en cuanto a contenido, esperando que poco a poco vayamos consiguiéndolo os deseamos desde aquí ¡Felices vacaciones!

Sumario:

TU OPINION NOS INTERESA ____ Pág. 4
TIPOGRAFIA CALAMUS ____ Pág. 6
ESCUCHANDO TU STE ____ Pág. 8
GUIA COMPLETA LENGUAJE C ____ Pág. 10
GRAFICOS: PERSPECTIVA CONICA ____ Pág. 13
INTERRUPCIONES EN EL ATARI (II) ____ Pág. 20
JUEGOS (PREVIEWS, ST, PC, LYNX,...) ____ Pág. 23
CODIGO MAQUINA ____ Pág. 32
HARDWARE: PSION ORGANISER II ____ Pág. 36
CARTAS Y CONTACTOS ____ Pág. 42

TU OPINION NOS INTERESA

Después de varios meses recopilando la opinión de nuestros lectores, vuestra opinión, hemos podido elaborar unas tablas de preferencias que unidas a un mayor esfuerzo por nuestra parte nos permitirán acercar la revista al ideal del lector dejando atrás el fantasma de las revistas extranjeras.

Vuestras opiniones se caracterizan por ser una sucesión de críticas constructivas que nos van a ayudar a enfocar la línea que debe seguir la revista. Como habréis podido apreciar ya hemos comenzado a cambiar, trabajando en nuevos artículos originales e independientes.

Las estadísticas de tu opinión nos interesa nos han revelado que en general las secciones que más interesaban hasta ahora son gráficos, lenguajes y noticias, seguidas de periféricos, hardware y juegos. Las menos interesantes fueron certámenes, comunicaciones y entrevistas, estando autoedición, música, cartas y contactos en lugares intermedios.

La mayor parte de nuestros lectores desea una revista en su mayor parte profesional, dedicando un 35% a juegos, un 47% a profesional y un 18% a otros. Curiosamente, los gustos varían con la edad, los más jóvenes dedican el 60% del tiempo de ordenador a juegos y un 20% a profesional, mientras que los más mayores dedican un 11% a juegos y un 67% a profesional, teniendo mayor peso alrededor de los 24 años, que es la edad media de nuestros lectores, que dedican el 37% a juegos y el 44% a usos profesionales.

Para intentar que la revista guste a todos los lectores, tenemos que estudiar las preferencias de cada uno por edades, teniendo en cuenta que el 14% tiene menos de 16 años, siendo el más joven de 9 años, el 45% tiene entre 16 y 25 años, el 32% tiene entre 26 y 35 años y el 9% tiene más de 35 años.

Por edades podemos decir que los más jóvenes han ordenado las secciones del siguiente modo, juegos, gráficos, noticias, contactos, entrevistas, cartas, certámenes, música, periféricos, hardware, autoedición, comunicaciones y lenguajes aunque curiosamente repartirían la revista dando 25% a juegos, 14% cartas, 10% lenguajes y 10% a noticias. Esto quiere decir que les gustan un tipo de secciones como lenguajes, aunque no les han gustado las publicadas hasta ahora.

El siguiente grupo, entre 16 y 25 años, ordena las secciones ya publicadas como sigue: lenguajes, gráficos, noticias, música, periféricos, hardware, contactos, juegos, cartas, autoedición, comunicaciones, certámenes y entrevistas pero darían un 13% a lenguajes, un 12% a juegos y un 11% a gráficos.

El grupo entre 26 y 35 años ha ordenado las secciones como sigue: lenguajes, gráficos, noticias, periféricos, hardware, cartas, autoedición, música, juegos, contactos, comunicaciones, certámenes y entrevistas; y en un futuro dedicarían un 12% a lenguajes, un 11% a gráficos, un 10% a juegos y un 10% a música.

El último grupo, de más de 35 años, ordenaron las secciones del siguiente modo: gráficos, autoedición, noticias, hardware, música, lenguajes, comunicaciones, periféricos, juegos, cartas, certámenes, entrevistas y contactos.

No obstante este grupo también da mayor importancia para el futuro a los lenguajes recomendándonos que dediquemos un 17% a lenguajes además de un 15% a noticias, un 11% a gráficos, un 11% a hardware, un 10% a música y un 10% a certámenes.

Se puede apreciar, que con la edad cada vez interesan menos los juegos, cartas y contactos mientras que el interés por lenguajes y gráficos es una constante. El interés por periféricos y hardware va a lo largo de los años siendo cada vez un tema más interesante.

Tenemos que destacar, que el único grupo que desea dedicar la mayor parte del tiempo a juegos son los más jóvenes, que pronto, dado que su edad media oscila alrededor de los 14 años, pertenecerán a un grupo de usuarios que irá adquiriendo interés por programación y usos cada vez más profesionales, suponemos que comenzarán con la autoedición, ya que permite a cualquier estudiante de BUP, FP y Universidad entregar trabajos con presentación inmejorable.

Hay un dato de las estadísticas que resulta escandaloso, el 99.2% de las encuestas que hemos recibido han sido rellenadas por varones mientras el resto, el 0.8 han sido remitidas por mujeres. Aquí en la redacción de C.B.C. Press, creíamos que la época en que a las niñas se las «castigaba» a jugar exclusivamente con muñecas y cocinitas ya estaba superada, teniendo por tanto sentido que las niñas, chicas y mujeres de hoy se interesen también en temas relacionados con la informática ya que esta puede ayudar profesionalmente o simplemente para llenar tiempo de ocio (juegos) a ambos sexos. Confiamos que esto no suceda en futuras entrevistas que podamos realizar.

Todos los lectores, desde un punto de vista global, coinciden por lo general en que la revista dedique espacio a comentar periféricos que aún no siendo de ATARI, sirvan para éste.

Por último, una vez vistos los datos numéricos, queda comentar el apartado de sugerencias. Apartado, en el cual los lectores comparan una y otra vez la revista ATARI USER con revistas extranjeras, dando por hecho que una revista extranjera es un modelo ideal a seguir.

Nosotros, estamos convencidos de que un mayor trabajo por nuestra parte y una mayor colaboración por la vuestra, pueden hacer que nuestra revista sea una de las más prestigiosas de Europa de la cual puedan copiar todos. La mayor parte de la revista, casi el 100%, se escribe enteramente en España, siendo por tanto todos los artículos españoles, no habiéndose publicado anteriormente en ningún otro sitio.

Prácticamente todos los lectores nos piden que profundicemos más en los artículos, sugerencia de la cual hemos tomado nota, y para ello hemos solicitado ayuda a Ordenadores Atari, S.A. y al departamento de desarrollo de Ibercomp, que nos dan soporte técnico para poder crear artículos con una mayor precisión y profundidad. Este tema es algo que ya habréis podido notar.

Bastantes usuarios nos piden una revista a todo color, con más páginas y con disco, algo que hoy no es posible todavía. Por ahora sólo podemos subir la calidad de la revista y aumentar su contenido sin aumentar el número de páginas.

Nosotros creemos, y vosotros nos dais la razón, que lo más importante de ATARI USER es el contenido y no el grosor. De todas formas, haremos lo posible para que, en un futuro no demasiado lejano, la revista sea a colores, con más páginas y disco.

Muchos nos sugerís cursos de programación, especificando lenguajes BASIC y 68000. Nosotros, después de haber hablado con expertos, hemos decidido dedicarnos en la mayor parte a lenguajes C y 68000. ¿Por qué? El ensamblador 68000 simplemente porque lo piden un buen número de lectores. Por otro lado, el sistema operativo del ATARI ST/TT está escrito casi totalmente en lenguaje C, siendo éste además el lenguaje en el que están escritas la mayoría de programas serios del ATARI, por ejemplo el CALAMUS está escrito en Turbo C para Atari. El lenguaje C es unas 20 a 100 veces más rápido que el BASIC compilado y no es más difícil escribir un programa en C que en Basic. Probablemente sí sea más complicado aprender C que aprender Basic, dado que para programar en C hay que saber exactamente lo que es un ordenador.

TU OPINION NOS INTERESA

ATARIUS.PRG

TU OPINION NOS INTERESA

Nombre:José García Lopez Provincia:8

Ordenador:1 Lo tengo desde:5 Edad: 16 Ocupacion:1

Entrevistas	:8	Entrevistas	:31
Gráficos	:3	Gráficos	:10
Autoedición	:13	Autoedición	:0
Juegos	:1	Juegos	:31
Noticias	:11	Noticias	:5
Música	:2	Música	:20
Comunicaciones	:12	Comunicaciones	:21
Contactos	:10	Contactos	:2
Cartas	:9	Cartas	:3
Periféricos	:6	Periféricos	:5
Hardware	:5	Hardware	:10
Certámenes	:7	Certámenes	:5
Lenguajes	:4	Lenguajes	:5

INTERES POR LA REVISTA

NADA POCO MUCHO

JUEGOS :90
PROFESIONAL:10
OTROS :0

NUEVO IR 001
GRABAR EST. 1
CARGAR EST. 2
SALIR EST. 3

Nosotros intentaremos escribir una serie de artículos que expliquen detalladamente cómo conseguir programar perfectamente en lenguaje C o lenguaje ensamblador partiendo de cero, esto es que tanto un curso como otro pueda ser seguido por cualquier lector aunque no sepa nada de programación. Somos conscientes de que existe un buen número de lectores que ya sabe programar, para los cuales dedicaremos artículos de programación más avanzados como la serie dedicada a interrupciones o secciones que dedicaremos a efectos especiales, matemáticas, electrónica,...

También nos pedís, aunque en número más reducido, que dediquemos páginas a hardware que puedan realizar los propios lectores. A partir de ahora daremos bastante importancia a este apartado, aunque nos tendréis que disculpar si no es una sección fija y constante. Desarrollar hardware, aunque empecemos desde puntos básicos como añadir una pausa al ordenador o controlar reles desde él mismo lleva bastante tiempo ya que el departamento de desarrollo de Ibercomp, que colabora desinteresadamente con ATARI USER, debe probar una y otra vez los pequeños aditamentos de hardware que desarrollan para vosotros, al mismo tiempo de buscar materiales económicos, fiables y "fáciles" de encontrar.

Intentaremos, a partir de ahora comentar dentro de lo posible todo tipo de programas y libros que estén relacionados con ATARI que existan en España y fuera de ésta, así como el modo de conseguirlos (siempre que sea posible). Para ello pedimos a todas las casas de software y editoriales que puedan leer estas líneas, que por favor se pongan en contacto con nosotros o nos envíen sus productos.

A pesar de que hasta ahora no os han gustado las secciones que mencionan entrevistas y certámenes que suceden de tanto en cuanto, intentaremos a partir de ahora darles otro enfoque que los haga más interesantes.

Respecto a la sección de comunicaciones, es posible que dediquemos de vez en cuando algún artículo, para que no perdáis la pista, y de un modo práctico, ya sea mediante listados que permitan comunicarse a unos ordenadores con otros (rutinas XMODEM, algún que otro SERVER para red local, etc) o bien con artículos de hardware, que permitan construir nuestro propio MODEM o bien aditamentos que permitan comunicar ordenadores.

Por otra parte algunos lectores nos piden que hablemos del ATARI XL, y consolas de videojuegos. Nosotros en ningún momento hemos tenido interés en no publicar artículos relacionados con dichos productos. Sucede que ATARI USER es una revista enfocada a todos los usuarios de ordenadores ATARI, usuarios que van desde los míticos ATARI 400 y 800 hasta los potentísimos Transputer, pasando por STFM, STE, TT, Consolas, [...] e impresoras intentando contentar al mayor número de usuarios. Dado que la mayoría tienen ordenadores ST y STE, la mayor parte de los artículos deben ser dedicados a estos dos ordenadores, aunque de vez en cuando se escriban artículos exclusivos para el TT, Transputer, XE y XL. Aún así, intentaremos llevar un orden sobre los artículos publicados para los ATARI «minoritarios».

Por último, existen tres secciones, que no dependen de nosotros, que son la sección de «contactos», que debe ser rellenada íntegramente por los lectores y cuya extensión depende de la cantidad de contactos que nos lleguen, la sección de cartas que depende asimismo de las que nos enviéis, a mayor número de cartas mayor número de respuestas, y la sección (entendida como guía de distribuidores) de publicidad, algo ajeno a nosotros y que depende del deseo de las distintas casas de llegar a vosotros a través de ATARI USER.

No obstante, vosotros los lectores, si lo deseáis siempre podéis ayudarnos a mejorar la revista, no sólo enviándonos contactos o escribiendo cartas preguntándonos dudas, sino que además os abrimos las páginas para que publiquéis vuestros artículos, programas y en definitiva cualquier cosa que se os ocurra, que esté relacionada con ATARI y pueda ser de interés para los demás lectores.

Programa escrito en Láser C específicamente para realizar la estadística de ATARI USER «Tu opinión nos interesa». 26 Kbytes de programa compilado más 8 Kbytes de recursos editados con el WERCS de Hisoft.

• Distribución por edades

- de 9 a 15 años = 14%

- de 16 a 25 años = 45%

- de 26 a 35 años = 32%

- de 35 a 49 años = 9%

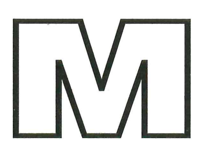
• Edad media = 24 años

• Tiempo medio de posesión del ordenador = 19 meses

Este artículo, es una buena prueba de lo que podéis hacer vosotros, ha sido realizado con vuestra ayuda y vuestro interés y toda la redacción de ATARI USER y colaboradores os damos las gracias por ayudarnos a mejorar.

PROTECCION DE LA TIPOGRAFIA DEL CALAMUS

Por
R. Lucas Rotger

uchos usuarios de A-TARI, utilizan el A-TARI fundamentalmente como instrumento de autoedición para autocrearse todo tipo de documentos y formularios. En los programas de autoedición, hoy por hoy, el Calamus es el programa que tiene una relación calidad-precio más alta, ya que a pesar de su precio elevado (alrededor de unas 60.000 pesetas en su versión monocroma) ofrece una calidad a la hora de imprimir superior a cualquier programa de la competencia, ya sea PC (ej.: Ventura, Page Maker, ...), ST (Timeworks, Pagestream, ...) o MAC (Aldus PageMaker, Quark Xpress, ...).

Una de las particularidades del Calamus es que utiliza Fonts vectoriales, tanto por pantalla como por impresora, obteniendo así una calidad perfecta al desaparecer los escalones típicos de otros programas similares que utilizan técnicas raster.

Para el Calamus existen una gran cantidad de tipos de letras, se cuentan a cientos. Existen dos clases de tipos de letras, los tipos de letra comerciales y los tipos de letra de dominio público. La diferencia estriba en que los tipos de letra comerciales sólo se pueden cargar desde el Calamus al cual están asociados por número de serie.

Cualquiera que disponga de un Calamus original, podrá ir con el ratón a la opción CALAMUS DMC y ver su número de serie. Con este número de serie, se puede solicitar a TOU (Distribuidor exclusivo del Calamus) tipos de letra. Estos no podrán ser utilizados por otro usuario que disponga de un Calamus con otro número de serie.

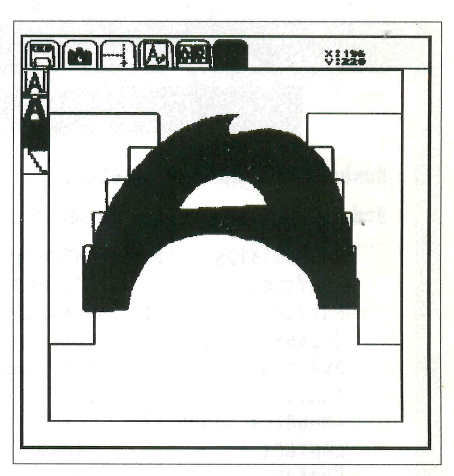
Para conseguir esto cada tipo de letra del Calamus dispone de un número de serie que debe coincidir con el del Calamus con el que se va a utilizar, o bien ser 0.

Una de las ventajas del Calamus, es que con ayuda de un accesorio se pueden editar o crear tipos de letra para el Calamus. El problema está en que este editor llamado FONTEDACC sólo permite editar/crear tipos de letra de dominio público, esto es con número de serie 0.

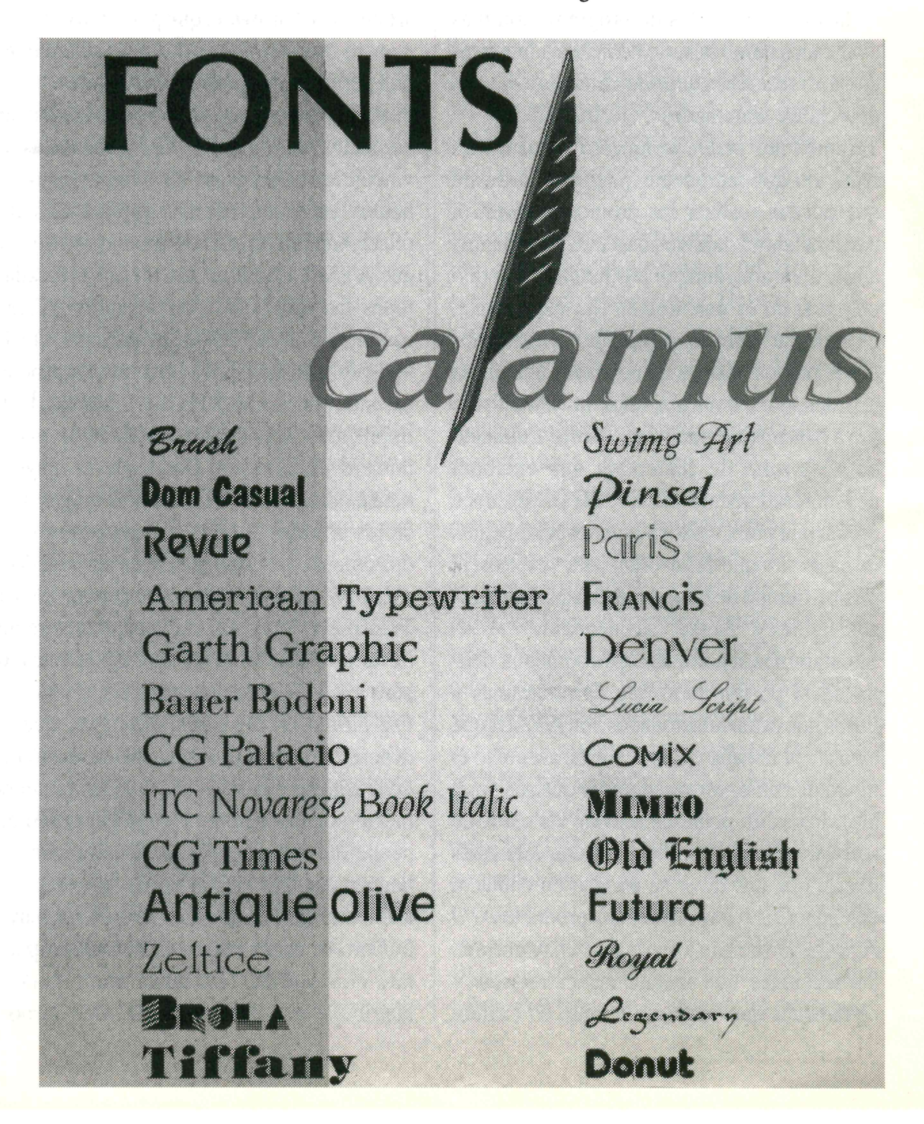
Para ayudar a aquellos lectores que deseen editarse un tipo de letra protegido, o crear un tipo de letra protegido para así poderlo comercializar, publicamos un pequeño listado en BASIC (HISOFT BASIC) que permite cambiar el número de serie a los fonts del Calamus.

Este programa nos pedirá el directorio donde se encuentra el tipo de letra a cambiar el número de serie, que puede ser D:\CALAMUS\LETRAS\. A continuación nos pide el fichero que deseamos modificar, este puede ser TIME50.CFN o bien *.CFN si deseamos modificar todos. Por último el programa nos pide el número de serie que deseamos, que será el de nuestro Calamus o bien -3325 si deseamos que sea cero.

Esperamos que este pequeño programita ayude a más de un lector, aunque recordamos que piratear no es un acto legal, nuestra intención es que podáis adaptaros los tipos de letra a vuestra medida, no que hagáis colección ilegal.



Pantalla del Editor de fonts



```
'           (C) 1991 Ibercomp SRL - ATARI USER/CBC Press
'                      escrito por: R. Lucas Rotger

library"gemdos"

dim dta%(22)

print"*** ATENCION: Si este programa no es usado correctamente, el
autor"
print"                  no se responsabiliza de ningun daño causado a los'
print"                  discos a que el programa acceda."
print"*** Este programa funciona correctamente, siempre que se sepa"
print"    manejar."
print string$(70,"—")
print"Camino: [Unidad de disco:\][Directorio\...]"
print"Archivo: <Nombre.Extension>"
print"En el archivo se pueden especificar caracteres comodin"
print"Para salir, responder con <Return> a las preguntas"
input"Camino:";camino$
input"Archivo:";arch$
pathstr$=camino$+arch$
if pathstr$="" then goto fin
no_vale:
input"Version para proteger";version&
if version&=0 then version&=10000
if version&=-3325 then version&=0
if len(hex$(version&))>6 then
  print"De 0 hasta 16777215 solamente"
  goto no_vale
end if
numerito$=string$(6-len(hex$(version&)),"0")+hex$(version&)
call dir2str(pathstr$,targstr$)
if targstr$=chr$(0) then print:print"No se han encontrado
archivos":goto fin
puntero=1
archivo$=""
while puntero<=len(targstr$)
  while mid$(targstr$,puntero,1)<>chr$(0)
    archivo$=archivo$+mid$(targstr$,puntero,1)
    puntero=puntero+1
  wend
  print"Modificando ";archivo$
  open camino$+archivo$ for random as 1 len=4
  field 1,4 as cccp$
  get 1,5
  mid$(cccp$,2)=chr$(val("&h"+mid$(numerito$,1,2)))
  mid$(cccp$,3)=chr$(val("&h"+mid$(numerito$,3,2)))
  mid$(cccp$,4)=chr$(val("&h"+mid$(numerito$,5,2)))
  put 1,5
  close 1
  puntero=puntero+1
  archivo$=""
wend

fin:
system

sub dir2str(pathstr$,targstr$)
  static isitthere,olddta&,addr&,hold
  redim dta%(22)
  olddta&=FNfgetdta&
  fsetdta varptr(dta%(0))
  isitthere=FNfsfirst%(pathstr$,0)
  if isitthere<0 then
    targstr$=chr$(0)
    fsetdta olddta&
    exit sub
  end if
  do
    addr&=varptr(dta%(15))
    do
      hold=peekb(addr&)
      if hold=0 then
        targstr$=targstr$+chr$(0)
        exit loop
      end if
      targstr$=targstr$+chr$(hold)
      incr addr&
    loop
    isitthere=FNfsnext%
    if isitthere<0 then
      fsetdta olddta&
      exit loop
    end if
  loop
end sub
```

Listado 1, protección de tipos de letras del CALAMUS.

R. Lucas Rotger

FILMADORA DTC 3000

La DTC 3000 es una filmadora que ofrece, junto con el equipo de autoedición de Atari, unas prestaciones muy altas a un precio verdaderamente sorprendente.

Pudiendo trabajar a resoluciones desde 480 hasta 3000 p.p.p. esta filmadora es líder en el mercado en cuanto a su relación calidad-precio. Utiliza la más avanzada tecnología de diodo láser para generar una imagen insuperable.

Para su funcionamiento se utiliza el equipo de autoedición Atari con el programa Calamus, sin ningún tipo de RIP ni interface especial, ya que se conecta directamente con el ordenador (modelo Atari TT030). De este modo se garantiza una total fiabilidad en cuanto a resultados, y una velocidad de filmación increíble (un minuto para un DIN-A4 a 1200 p.p.p.). La fabricación de la filmadora se realiza en una empresa del grupo SIEMENS Ag, con lo que la calidad de fabricación está garantizada.

COMPUGRAPHIC LINOTRONIC

Atari también está preparado para conectarse a otros sistemas de filmación.

El principal sistema de conexión a este tipo de equipos, reside en el elevado coste que suponen el interface entre el ordenador y la filmadora (el RIP).

Esta desventaja ha sido superada por los sistemas de autoedición Atari, al haber desarrollado la firma alemana, DMC, un interface de bajo coste capaz de manejar tales filmadoras.

El modelo de interface para las filmadoras Linotronic es capaz de controlas los modelo 100, 200, 300, 500 y 330, pudiendo filmar una hoja a 1270 puntos en formato DIN A4 en 3 ó 4 minutos.

Por otro lado, en lo que respecta a la conexión a los modelos de Compugraphic, también se ha desarrollado otro interface capaz de conectarse a los sistemas 9400 y 9600, con una velocidad de filmación muy similar a la que se consigue con las filmadoras Linotronic.

ESCUCHANDO TU STE

Cómo hacer sonar al LMC 1992

Por
David Cortés Provencio

Voy a contaros algo del nuevo chip de sonido del STE; básicamente se trata de un convertidor digital/analógico (D/A en lo que sigue), del tipo PCM (pulse code modulation, algo así como modulación por código pulsado) con una resolución de 8 bits y 4 velocidades de muestreo (aproximadamente 50 KHz, 25 KHz, 12.5 KHz y 6.25 KHz), con un filtro pasa-bajos para producir el efecto "antialiasing" de los datos sonoros además de otro pasa-bajos a 16KHz y por fin un chip LMC 1992 controlador de volumen y tono fabricado por National.

Todo esto sale por las clavijas de tipo RCA que tenéis en vuestro Atari en estéreo y mezclado con el sonido del antiguo chip (por cierto, se puede controlar gracias al LMC 1992 el balance de la mezcla).

Supongo que todos estos datos técnicos os dirán poco si no estáis metidos en el mundo de la música y/o electrónica, intentaré aclararos alguna cosita:

Un convertidor D/A P.C.M. de 8 bits en nuestro caso lo que hace es transformar números binarios de 8 bits a desplazamientos del altavoz (concretamente el número binario +127 mide el mayor desplazamiento positivo y -128 el menor negativo, que son los límites de representación de un número binario en formato de complemento a dos), este desplazamiento del cono del altavoz a través del tiempo produce una oscilación que normalmente es audible, si esta variación es más o menos regular se oye un sonido, si es muy irregular o aleatoria se oye un ruido (como luego escucharéis). La velocidad de muestreo influye en el rango de frecuencias que se escucharán (por ejemplo hay más agudos a 50 KHz que a 6.23 KHz). Los controles de tono y volumen son análogos a los que tenéis en vuestros equipos HIFI o radiocassettes para manejo de agudos, graves y volumen (para los amantes de los datos diré que el control de tonos tiene una ganancia/atenuación de +/- 12 dB a 50 Hz y 15 KHz y que el volumen atenúa hasta -80 dB).

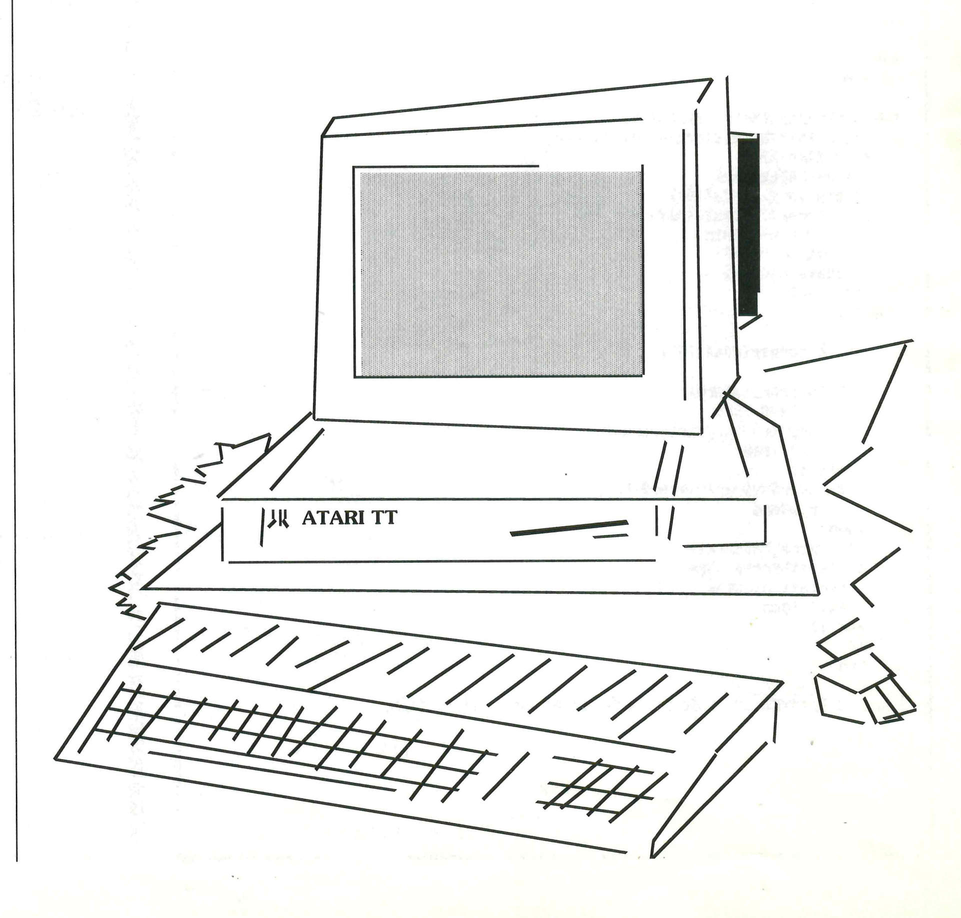
Bueno, creo que queda un poco más claro qué es el nuevo chip de sonido y para que lo podáis oír, aquí va un programa en ensamblador que lo único que hace es una conversión D/A de los números comprendidos entre la dirección $000000 y la dirección $000f00 que como podréis imaginar, son bastante aleatorios:

PRUEBA CHIP DMA (ver recuadro página siguiente)

Como no es mi objetivo, por el momento, enseñaros a programar en ensamblador y como tampoco tendría espacio para una explicación pormenorizada de las instrucciones que utilizo sólo voy a haceros una descripción por bloques general y me entretendré un poco más en lo concerniente al manejo del sonido.

El diagrama de bloques del programa podría ser:

PONER EN MODO SUPERVISOR AL 68000
:
:
IMPRIMIR INSTRUCCIONES DE USO
:
:
DEFINIR A1 COMO PUNTERO PARA EL MANEJO DE DATOS DEL CHIP (POR DIRECCIONAMIENTO INDIRECTO CON DESPLAZAMIENTO)
:
:
LEER TECLADO
:
:
ESTABLECER QUE TECLA SE PULSO Y OBRAR EN CONSECUENCIA



SONIDO

Como veréis en las instrucciones de uso se puede hacer lo siguiente:

- <u>Cambiar la frecuencia de muestreo</u>: esto se realiza accediendo a los bits 0 y 1 del byte 32 (a1) su dirección real es $FF8920, los cambios son:

00 - 6.25 KHz	10 - 25 KHz
01 - 12.5 KHz	11 - 50 KHz

- <u>Cambiar a estéreo o mono</u>: aunque en la documentación de Atari se especifica como válido el bit 8 de la dirección $FF8920 yo he encontrado como válido el bit 8 de la dirección $FF8921 (33 (a1) en lugar de 32 (a1)) si alguien encuentra el porqué que se ponga en contacto conmigo, funciona así:

0 - Estéreo	1 - Mono

- <u>Cambiar volumen, agudos y graves</u>: se accede a través de una especie de interface serie que se llama MICROWIRE y se hace funcionar poniendo un registro interno (del chip conectado a él) junto con su dato en microwire data y el código de acceso a ese chip en microwire mask (en nuestro caso es $07FF), por ejemplo:
(represento los 9 bits de menor peso de data)

instrucción__	dato_	
011	DDD DDD	Pone el Volumen
010	XXD DDD	Pone Agudos
001	XXD DDD	Pone Graves

bits: X - no importa el valor
D - dato en binario
(ganancia/atenuación de 2 dB por paso)

Sé que los menos avezados en estos menesteres tendrán la impresión de haber leído un cuento en otro idioma, pero el acceso real del nuevo chip es así, de todas formas en BASIC podréis acceder a las posiciones de memoria, siempre y cuando estéis en modo supervisor, sin necesidad de ensamblador.

Los que estéis interesados podéis acudir a la siguiente bibliografía:

- Extracto del manual de desarrollo de Atari
- Concise Atari 68000 programer´s reference guide, por K.D. Peel
- Microprocesadores de 16 bits, por José M. Angulo

Si tenéis alguna duda, ya sabéis que podéis escribir a la dirección de ATARI USER.

Hasta pronto.

```
**************************************
***      PRUEBA CHIP DMA         ******
*** (C) 1991 D.C.P. SOFT & MUSIC ******
**************************************
gemdos          equ     1
super           equ     $20
xbios           equ     14
c_conws         equ     9
c_rawcin        equ     7
*************************************

        clr.l   -(sp)
        move    #super,-(sp)
        trap    #gemdos
        addq.l  #6,sp
        move.l  d0,save_ssp
start
        move.l  #string,-(sp)
        move    #c_conws,-(sp)
        trap    #gemdos
        addq.l  #6,sp
        move.l  #$ff8900,a1        * puntero a: sound dma control
        move    #$00,2(a1)      * dir principio secuencia: 000000
        move    #$00,4(a1)
        move    #$00,6(a1)
        move    #$0000,8(a1)       * dir cont.
        move    #$0000,10(a1)
        move    #$0000,12(a1)
        move    #$00,14(a1)     * dir final sec.: 000f00
        move    #$0f,16(a1)
        move    #$00,18(a1)
        move    #$0000,32(a1)      * control: 6258 Hz y stereo
        move    #$04ef,34(a1)      * microwire data   (volumen max.)
        move    #$07ff,36(a1)      * microwire mask
        move    #3,(a1)            * ! una de ruido !
bucle1
        move    #c_rawcin,-(sp)  * leer teclado
        trap    #gemdos
        addq.l  #2,sp
        cmp.l   #$001c000d,d0      * <return>
        bne     salto1
        move    #0,(a1)
        bra     end
salto1
        cmp.l   #$001e0041,d0       * <A>
        bne     salto2
        and.b   #$f0,32(a1)
salto2
        cmp.l   #$001e0001,d0       * ctrl + <A>
        bne     salto3
        and.b   #$f0,32(a1)
        or.b    #$01,32(a1)
salto3
        cmp.l   #$001e0000,d0       * alt + <A>
        bne     salto4
        and.b   #$f0,32(a1)
        or.b    #$02,32(a1)
salto4
        cmp.l   #$00300042,d0       * <B>
        bne     salto5
        or.b    #$03,32(a1)
salto5
        cmp.l   #$00300002,d0       * ctrl + <B>
        bne     salto6
        and.b   #$0f,33(a1)
salto6
        cmp.l   #$00300000,d0       * alt + <B>
        bne     salto7
        or.b    #$80,33(a1)
salto7
        cmp.l   #$002e0003,d0       * ctrl + <C>
        bne     salto8
        move.l  #volumen,a3
        cmp     #40,(a3)
        beq     salto8
        add     #1,(a3)
        move    (a3),34(a1)
        or      #$04c0,34(a1)
        move    #$07ff,36(a1)
salto8
        cmp.l   #$002e0000,d0       * alt + <C>
        bne     salto9
        move.l  #volumen,a3
        cmp     #0,(a3)
        beq     salto9
        sub     #1,(a3)
        move    (a3),34(a1)
        or      #$04c0,34(a1)
        move    #$07ff,36(a1)
salto9
        cmp.l   #$00200004,d0       * ctrl +<D>
        bne     salto10
        move.l  #agudos,a3
        cmp     #12,(a3)
        beq     salto10
        add     #1,(a3)
        move    (a3),34(a1)
        or      #$0480,34(a1)
        move    #$07ff,36(a1)
salto10
        cmp.l   #$00200000,d0       * alt +<D>
        bne     salto11
        move.l  #agudos,a3
        cmp     #0,(a3)
        beq     salto11
        sub     #1,(a3)
        move    (a3),34(a1)
        or      #$0480,34(a1)
        move    #$07ff,36(a1)
salto11
        cmp.l   #$00120005,d0       * ctrl + <E>
        bne     salto12
        move.l  #graves,a3
        cmp     #12,(a3)
        beq     salto12
        add     #1,(a3)
        move    (a3),34(a1)
        or      #$0440,34(a1)
        move    #$07ff,36(a1)
salto12
        cmp.l   #$00120000,d0       * alt + <E>
        bne     salto13
        move.l  #graves,a3
        cmp     #0,(a3)
        beq     salto13
        sub     #1,(a3)
        move    (a3),34(a1)
        or      #$0440,34(a1)
        move    #$07ff,36(a1)
salto13
        bra     bucle1
end
        move.l  save_ssp,-(sp)
        move.w  #super,-(sp)
        trap    #gemdos
        add.l   #6,sp
        clr     -(sp)
        trap    #gemdos

string  dc.b  27,"E"
        dc.b  "              (C) 1.991 by D.C.P. Soft & Music ",13,10,10,
        dc.b  "                          MENU                 ",13,10
        dc.b  "                        ~~~~~~~~~~             ",13,10
        dc.b  "   TECLA:  NORMAL:      CON CTRL.:   CON ALT.: ",13,10
        dc.b  "  ======= =========     ============= ===========",13,10
        dc.b  "     <A>    6258 Hz     12517 Hz     25033 Hz  ",13,10
        dc.b  "     <B>   50066 Hz     Stereo       Mono      ",13,10
        dc.b  "     <C>   Volumen      Inc.         Dec.      ",13,10
        dc.b  "     <D>   Agudos       Inc.         Dec.      ",13,10
        dc.b  "     <E>   Graves       Inc.         Dec.      ",13,10
        dc.b  "  <Return>   FIN        -------      -------   ",13,10
        dc.b  10,13,10,13,10,13,10,13,10,13,10
        dc.b  "                 !PRECAUCION! SOLO MAYUSCULAS ",13,10
        dc.b  0,0
save_ssp        dc.l    0
tecla           dc.l    0
volumen         dc.w    40
agudos          dc.w    6
graves          dc.w    6
```

GUIA COMPLETA DEL LENGUAJE C (II)

En el número anterior de Atari User, en la primera parte de la Guía intentamos introduciros al lenguaje C, diferenciándolo con el Basic y otros lenguajes. Además también quisimos poneros al día de las variables existentes para este lenguaje ¿lo conseguimos? En este número 28 nos meteremos en el Flujo de programa, esperando que os resulte interesante, ponemos manos a la obra...

FLUJO DE PROGRAMA

Antes de entrar de lleno en la creación de flujos observaremos brevemente a los operadores lógicos, que son los mismos que hay en BASIC, pero se escriben de forma distinta.

Lenguaje BASIC	Lenguaje C
=	==
<>	!=
>	>
<	<
>=	>=
<=	<=
AND	&&
OR	||
NOT	!

Pequeñas diferencias, a las cuales el futuro programador en C se habituará rápidamente. En lenguaje C contamos con:
if, else, else if;
y un ejemplo ilustrativo de cómo se utiliza podría ser:

```
main ()
{
        int a, b, c;

        a=1;
        b=1;
        c=1;

        if (a==b) {
        c=a+b;
        }
        else if (a>b) {
        c=a;
        }
        else {
        c=b;
        }
```

Evaluación sencilla, que comprueba si a es igual a b, en caso afirmativo c=a+b, en caso negativo comprueba si a es mayor que b y c igual a a y en caso contrario (que sea menor) hace c=b.

Instrucción for, esta es bastante diferente a la que conocemos en BASIC, aunque puede simular más o menos a ésta.

Si deseamos hacer:

```
FOR X=1 TO 100
        A=A+X
NEXT X
```

en lenguaje C deberemos escribir:

```
for (x=1;x<=100; x=x+1) {
     a=a+x;
}
```

o sea una estructura tipo for (asignación, condición, incrementación) en BASIC sería algo como:

```
        ASIGNACION
INICIO
        IF CONDICION THEN
                .
                .
                .
             INCREMENTACION
             GOTO INICIO
        END IF
```

Parece complicado, pero no lo es, aunque se puede complicar, ya que es posible hacer for muy raros inimaginables en BASIC:

```
for (x=1, y=500;                 'Asignación.
x<=100 && y>x && y>100; 'Comproba-
ción.
x=x+1,                           'Incremento.
y=y-x) {

a=a+x;                           'Lo de dentro
y=y-a-x;
}
```

Aparentemente un lío, y eso que no está todo en una línea, «for (x=1, y=500; x<=100 && y>x && y>100; a=a+x, y=y-x)», que sería lo normal. Traduciendo este super for a BASIC clásico sería algo como:

```
10 REM Asignación.
20 LET X=1
30 LET Y=500
40 REM Condición.
50 IF X<=100 AND Y>X AND Y>100
THEN
60 REM Lo de dentro
70 LET A=A+X
80 LET Y=Y-A-X
90 REM Incremento.
100 LET X=X+1
110 LET Y=Y-X
120 GOTO 40
130 END IF
```

En el lenguaje C el for, al igual que en el BASIC es anidable, esto es unos for pueden estar dentro de otros, por ejemplo:

```
main ()
{
int a, b, c, d, e, f, n;
 n=0
  for (a=1;a<7;a++) {
   for (b=1;b<7;a++) {
    if (b!=a) {
     for (c=1;c<7;c++) {
      if (c!=b && c2=a) {
       for (d=1;d<7;d++) {
        if (d!=c && d!=b && d!=a) {
         for (e=1;e<7;e++) {
          if (e!=d && e!=c && e!=b && e!=a) {
          for (f=1;f<7;f++) {
          if (f!=e && f!=d && f!=c && f!=b && f!=a) {
         n++;
          }
          }
         }
        }
       }
      }
     }
    }
   }
  }
 }
printf («Resultado: %d\n\r»,n);
}
```

Al ejecutar este programa, aparecerá en pantalla "Resultado: 720», ya que después de este impresionante anidamiento de for e if, la variable n acaba tomando el valor 720.

De este ejemplo hay que sacar además dos cosas nuevas. Una expresión tipo x=x+1 se puede escribir simplemente como x++ y una expresión del tipo x=x-1 como x- -. El ++ o el - - significan incrementar o decrementar en una unidad la variable que está delante. Este incremento o decremento puede estar en medio de cualquier cálculo, ejemplo:

```
a=1;
x=3;

x=(x++) + (a++);
```

hará que al final del proceso x tome el valor 5, ya que x+a es cuatro y 4++ es cinco y a tomará después de la línea el valor 2. Para ser más exactos la línea anterior hace lo mismo que:

```
x=x+a;
x=x+1;
a=a+1;
```

Otra cosa nueva es que es posible sustituir una operación x=x+lo_que_ sea por una expresión tipo x+=lo_que_sea, de forma que x=x+a es lo mismo que x+=a, que es lo correcto. También existen otras operaciones de este tipo:

x=x+n; es lo mismo que x+=n.
x=x- n; es lo mismo que x-=n.
x=x*n; es lo mismo que x*=n.
x=x/ n; es lo mismo que x/=n.

La instrucción rara que aparece en el ejemplo anterior es printf («Resultado: %d\n\r», n). En realidad no es una instrucción sino una función definida en el sistema. Su utilización es demasiado compleja, permite imprimir en pantalla cualquier texto y/o valor de variable bajo cualquier formato, (además del manejo de impresora y modem, y opcionalmente para escribir datos en el disco). Su formato es siempre:

```
printf ("texto entrecomillado\n\r", p1, p2, p3, ...);
```

de modo que cuando escribamos entre el texto entrecomillado %d, querrá decir que en su lugar debe ser escrito el valor del parámetro.

Ejemplo:

```
a=1;
b=2;

printf ("EJEMPLO %d más %d es %d. \n\r", a, b, a+b);
```

con lo cual aparecerá por pantalla "EJEMPLO: 1más 2 es 3."

Haciendo un uso más profundo de las instrucciones for, podemos escribir el anidamiento de antes como:

```
main ()
{
/*Este programa hace exactamente lo mismo que el anterior la diferencia estriba en que se han introducido los if dentro de los for.*/
int a, b, c, d, e, f, n;
n=0
for (a=1;a<7;a++) {
for (b=1;b<7;b++) {
for (c=1;c<7 && b!=a;c++) {
for (d=1;d<7 && c!=b && c!=a;d++) {
for (e=1;e<7 && d!=c && d!=b && d!=a;e++) {
for (f=1;f<7 && e!=d && e!=c && e!=b && e!=a;f++) {
if (f!=e && f!=d && f!=c && f!=b && f!=a){
n++;
}
}
}
}
}
}
}
printf(«Resultado: %d\n\r»,n);
}
```

Explicándolo de otra forma, cuando escribimos un for de la forma:

```
for (x=1;x<=10 && x!=b;x++, b=b-x){
lo_de_dentro ();
}
```

quiere decir que x tomará el valor 1, a continuación mirará si es menor o igual que 10 (x<=10), y si es distinto de b (&& x!=b) en caso afirmativo ejecuta lo_de_dentro (), y al final hace x=x+1 (x++) y b=b-x.

Tal y como se puede observar se trata de un for muy completo y muy útil si se utiliza bien. Lo que no debe hacerse o al menos evitarse es meter dentro del for comprobaciones que permanecen constantes a lo largo de todo el proceso for, ya que ralentiza el for debido a forzarlo a realizar la comprobación más veces (una por vuelta). Ejemplo:

```
for (x=1;x<=10000 && a==b;x++) {
n++;
}
```

si por algún motivo a y b fuesen iguales en el for se incrementaría n 10000 veces, se comprobaría si x<=10000 otras 10000 veces y si a==b otras tantas y se miraría si ambas condiciones son ciertas otras 10000 veces (el &&). Es obvio que si la primera vez a y b son iguales lo serán durante las 9999 restantes, con lo cual lo lógico es:

```
if (a==b) {
    for (x=1;x<=10000;x++) {
        n++;
    }
}
```

con ello ahorramos al programa 9999 a==b y 10000 &&, con el consiguiente ahorro de tiempo.

Para escribir cosas como estas el C ha previsto un híbrido entre for e if que se escribe del modo siguiente:

```
if (a==b) for (x=1;x<=10000;x++) {
n++;
}
```

Al estar programado en lenguaje C debe ser siempre el programador el que elija cual es el tipo de for que más conviene, y siempre hay que tener en cuenta, que lo que se busca es realizar el mínimo número de operaciones, no el ahorrar memoria (el ST tiene mucha) o escribir un programa con menos líneas.

Hasta aquí hemos visto con claridad el uso de las instrucciones de flujo «if», «for» e «if for». la siguiente instrucción de flujo que posee el lenguaje C es while. Su sintaxis es while (condición), por ejemplo:

```
x=10;
while (x>1) {
printf («La variable x vale %d\n\r»,x);
x- -;
}
```

es muy similar a un if, sólo que es como si llevase un GOTO a si mismo, o sea:

```
x=10;
inicio:
if (x>1) {
  printf («La variable x vale %d\n\r»,x);
  x- -;
  goto inicio;
}
```

eso sí, no tiene else, aunque a veces puede resultar útil, como en esta cuenta atrás. Aquí digo como antes, no hay que meter condiciones obvias, que no cambien dentro del while, para ello tenemos el híbrido.

```
a=3;
x=10000;
if (a==3) while (x>1) {
printf («La variable x vale %d\n\r»,x);
x- -;
}
```

de este modo nos ahorramos comprobar un par de miles de veces si a es igual a tres.

La siguiente instrucción es muy parecida a while, prácticamente su diferencia es nula, se trata de do, cuya sintaxis es do (...) while (condición), ejemplo:

```
x=10;
do {
   printf («La variable x vale %d\n\r»,x);
   x—;
} while (x!=0);
```

su equivalente es:

```
   x=10;
inicio:
   printf («La variable x vale %d\n\r»,x);
   x—;
   if (x!=0) {
        goto inicio
}
```

La siguiente instrucción de flujo es switch, que es idéntica a la instrucción SELECT del BASIC. Se utiliza de la siguiente forma:

```
switch (x) {
     case 1:
          ejemplo ();
          c=suma (a,x);
          break;
     case 3:
     case 4:
          x+=c;
          break;
     default:
          x=(x/2)*2;
          for (,x>0 && x % 2==0, x—) {
               c+=x;
          }
          break;
}
```

en este sencillo ejemplo, si x=1 llama a la función ejemplo () y asigna a c la suma de a y x. En caso de que x sea 3 ó 4 have x+=c. Y en cualquier otro caso (default:) hace x=(x/2) y el for. Explicándolo de otro modo se ejecuta aquella porción del programa que esté después del case n, donde n debe ser igual a x. La porción de programa a ejecutar debe terminar con un break. Sino se pone break, el proceso seguirá ejecutándose hacia adelante sin tener en cuenta la existencia de otras sentencias case.

En este ejemplo hay una operación nueva que es %, ésta devuelve el resto de la división, por ejemplo 5 % 2 es igual a 1, 34 % 7 es igual a 6. En algunos BASIC, como el superBASIC, a esta función se la denomina MOD, o sea PRINT 5 MOD 2 en un QL devuelve 1.

La última instrucción de flujo que posee el C es break, que es equivalente a EXIT en BASIC. Su único inconveniente, que sólo permite salir del switch, do, while o función más cercano. Ejemplos:

```
for (x=1;x<=100;x++) {
     if (x = = a) {
          break;
     }
}
```

este for, será interrumpido cuando x tome el valor de a. El problema viene cuando el programa está dentro de un for y se desea salir de varios for y el subprograma. En BASIC sería algo del estilo:

```
SUB CALCULO
 FOR X=1 TO 100
  FOR Y=1 TO 60
   IF X-Y=41 THEN
    EXIT SUB
   END IF
  NEXT Y
 NEXT X
END SUB
```

pero en C sería algo más complicado, aunque mucho más rápido, ya que es el programador el encargado de especificar por donde a de ir el flujo del programa, y no el RUNTIME.

```
calculo ()
{
  int x,y;
  int variable_de_apoyo;

  variable_de_apoyo=1;
  for ( x = 1; x < = 100 && variable _ de _
apoyo; x++) {
     for (y=1;y<=60;y++) {
       if (x-y==41) {
         variable_de_apoyo=0;
         break;
       }
     }
   }
}
```

Este cálculo se realizará como el anterior escrito en BASIC, sólo que aquí es el programador el que se encarga de hacer los EXIT. El for (x=1;x<=100 && variable_de_apoyo;x++) es como un for normal, con la particularidad que nunca será ejecutado si variable_de_apoyo vale 0. Así cuando x-1 es igual a 41 se sale del segundo for con ayuda del break, y del primero porque la variable_de_apoyo es 0.

Hay otras muchas formas, todas válidas y otra más, el programador es quien elije el modo más adecuado para cada porción de programa.

Otro ejemplo de lo mismo es:

```
Calculo ()
{
  int x,y;
  int variable_apoyo;

  variable_apoyo=1;
 for (x=1;x<=100;if (variable_apoyo) x++
else break){
    for (x=1;y<=60;y++) {
     if (x-y==41) {
       variable_apoyo=0;
       break;
     }
    }
  }
}
```

la ventaja de este segundo es que la comprobación se hace al final del for y no al principio. (Que es similar a donde comprueba un RUNTIME).

Aunque en C artimañas de este tipo siempre son necesarias para salir de un sitio, es cuestión de acostumbrarse.

Lenguaje C

Lenguaje C

Lenguaje C

Lenguaje C

Lenguaje C

Lenguaje C

Por A.Miguel

Algunos de vosotros habréis visto como muchos programas representan sus objetos vectoriales (sin utilizar Sprites) en 3 dimensiones sobre nuestra pantalla. Los programas pueden ser muy dispares, programas como el CAD 3D, Carrier Command, Starglider o cualquier otro, pero todos parten de las mismas técnicas geométricas que están basadas en la observación del mundo real.

INTRODUCCION A LA PERSPECTIVA

En este artículo, vamos a intentar acercar a los lectores los principios de la Perspectiva Cónica. Para comenzar sugerimos que el lector mire cada vez que se nombre un concepto nuevo la figura 1 y si es necesario que dibuje y realice bocetos a fin de «absorver» las ideas lo más rápidamente posible.

En primer lugar vamos a destacar en el espacio algunos elementos significativos, para ello vamos a suponer que estamos de pie mirando hacia el horizonte a la orilla del mar o en el campo. Así podemos inmediatamente darnos cuenta de que bajo nuestros pies hay un suelo, que no es más que una superficie plana, que imaginariamente se puede suponer infinita (hay tanto como queramos) y a la que se denomina Plano Geometral (PG).

Una vez que tenemos definido el plano geometral podemos definir otro paralelo a éste que pase por nuestros ojos, representados en la figura 1 como el punto de mira V. Este plano se denomina el Plano de Horizonte (PH).

Para representar objetos en tres dimensiones en dos dimensiones, esto es sobre papel o monitor, necesitamos un plano donde representarlos, este plano es el Plano del Cuadro (C-π) y en nuestro artículo será el monitor de nuestro ordenador o mejor dicho, el monitor será una porción del plano del cuadro. El plano del cuadro, normalmente está ante nuestros ojos y es perpendicular (forma 90 grados) con el Plano Geometral y Plano de Horizonte.

Por último tenemos otro plano, que es paralelo al plano de cuadro y pasa por nuestros ojos, a este plano se le conoce como el plano de desvanecimiento (ô). El nombre de desvanecimiento no es casualidad, sino que está dado por el hecho de que cualquier punto o figura que esté sobre este plano no podemos verlo ni representarlo ya que como veremos más adelante no es posible proyectar sus puntos sobre el plano del cuadro.

Existen además de los cuatro planos mencionados anteriormente dos rectas con nombre propio, la línea de horizonte (LH) que está donde se cortan el plano de horizonte y el plano del cuadro. La otra línea se la denomina línea de tierra (LT) y está donde se cortan el plano del cuadro y el plano geometral (suelo del monitor).

Para representar cualquier punto (M), debemos conocer sus coordenadas. Una vez conocidas, trazamos una recta imaginaria desde (M) (el punto) hasta V (nuestros ojos) y buscamos en qué punto la recta corta al plano del cuadro. Este punto, en nuestro ejemplo de la figura 1 es el punto M y como es lógico está situado sobre el plano del cuadro. Podemos también hacer lo mismo con el punto (M') que es el punto que está justamente debajo del punto (M) solamente que apoyado en el suelo. Su representación en el plano del cuadro sería el punto M'.

Si en este ejemplo hubiéramos tenido que representar un poste que apoyado en el suelo sobre (M') tuviese una altura h, este poste en nuestro monitor aparecería desde M' a M.

Hasta ahora nos hemos limitado a exponer un poco las ideas llamando a cada cosa por su nombre, pero sin asociarlas demasiado a la programación, aunque cualquier lector avanzado, con lo expuesto hasta aquí, ya está preparado para representar objetos en 3 dimensiones.

No obstante, nosotros vamos a desarrollar un pequeño programa en lenguaje C que representará un sencillo objeto en 3 dimensiones y permitirá que nos movamos alrededor de él.

Para ello debemos de buscar una serie de fórmulas que nos permitan a partir de las coordenadas del objeto y la descripción del entorno obtener las coordenadas en dos dimensiones de dicho objeto. En este artículo vamos a dar por hecho que el lector sabe programar en algún lenguaje (aunque nuestro ejemplo sea en lenguaje C a fin de que pueda complementar en cierto modo nuestros cursos de programación en este lenguaje), y que conoce alguna función que le permita dibujar una línea y posteriormente borrarla.

Pero antes debemos de conocer, aunque sea a modo de curiosidad, el concepto de Cono Optico. Si nosotros, los humanos, miramos hacia el frente sólo podremos ver los objetos que estén dentro de un cono de 60 grados cuyo vértice está situado en nuestro ojo y su eje es perpendicular a nuestra cara, o sea,

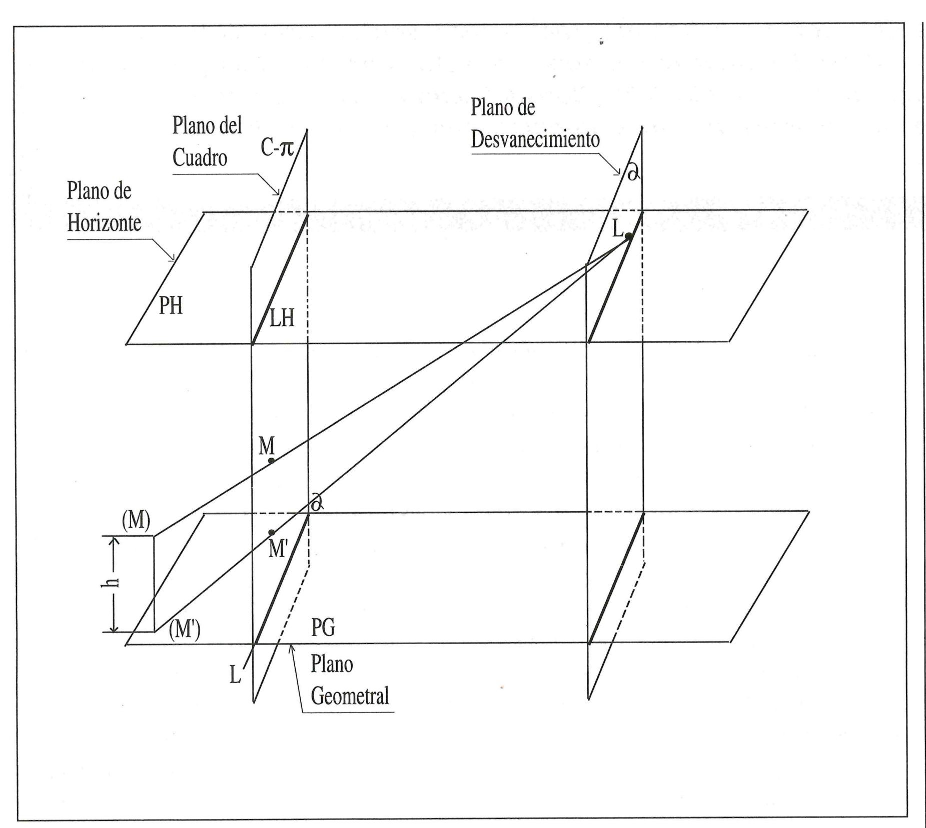


FIGURA 1

sale de nuestro ojo hacia adelante.

Otros seres vivos, como peces, aves, reptiles, etc., tienen conos ópticos diferentes como por ejemplo los 120 grados de un pez. Con un ángulo óptico de 120 grados la perspectiva es totalmente diferentes y es la conocida por todos y denominada ojo de pez, que aparece muchas veces en documentales de televisión para permitirnos ver paisajes o interiores.

El problema del cono óptico, se puede dividir en dos problemas diferentes, un problema vertical y un problema horizontal. Esto quiere decir que podemos usar el problema vertical para hallar la posición vertical de un punto en el plano del cuadro y el problema horizontal para hallar la posición horizontal en el plano del cuadro. De este modo simplificamos el problema de tres dimensiones en dos problemas de dos dimensiones que son en realidad el mismo. (Ver figura 2).

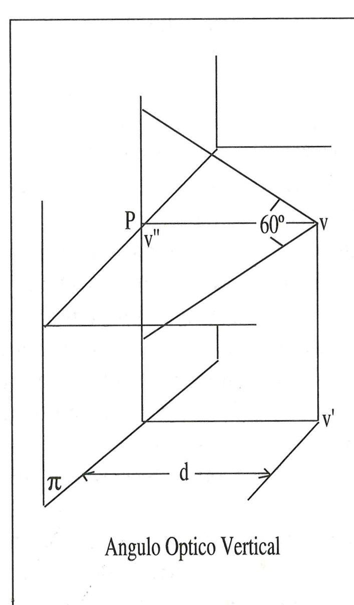
Antes hemos hablado de que hay que definir las variables de contorno, ya que de ellas va a depender el resultado final de nuestra representación, por ejemplo cambiar la altura del plano de horizonte implica una representación distinta de los objetos, algo que se puede apreciar en el cubo de la figura 3.

Para ello en primer lugar hay que definir unos ejes de coordenadas, para el mundo en tres dimensiones vamos a definir los ejes x - y - z y otro para situar los puntos dentro del plano del cuadro, los ejes cx-cy (ver figura 4). OJO: los cy-cx y z-y no tienen nada que ver, su paralelismo en el dibujo es pura coincidencia.

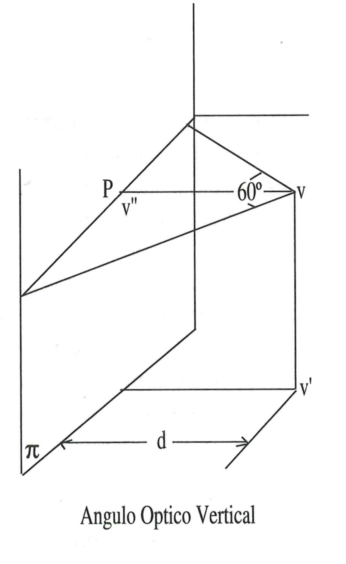
Otras variables que define el sistema son las coordenadas del punto de mira V en los ejes x-y-z, que determinan la posición de nuestra cabeza. Una vez determinada la posición de nuestra cabeza, debemos elegir el tipo de óptica que deseamos esto es si queremos un cono óptico de 60 grados, 120 grados o simplemente 10 grados. Esta variable es la que hace que la perspectiva sea más fuerte o menos. Esta variable está íntimamente relacionada con la distancia entre el punto V y el plano del cuadro. Si deseamos una óptica inicial de 60 grados y sabemos que el monitor tiene una resolución de 640 x 400 - siempre trabajaremos con monitores ST en nuestro análisis aunque el lector puede adaptarse la resolución a sus necesidades, pero teniendo en cuenta que las resoluciones media ST y media TT deforman las imágenes verticalmente y horizontalmente- debemos forzar a que toda la pantalla entre en el cono óptico, para ello:

$$d = \sqrt{\frac{3 \times (640^2+400^2)}{4}} = 654 \text{ puntos}$$

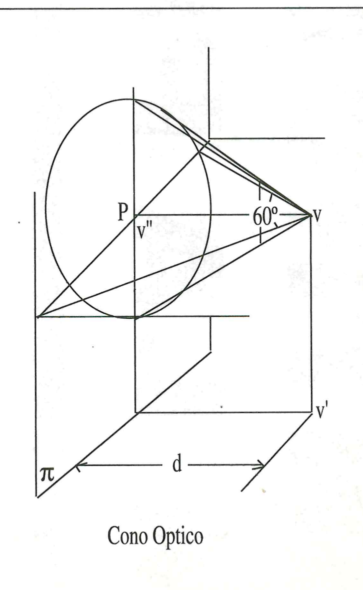
FIGURA 2
Para la representación de figuras en perspectiva cónica, lo más sencillo es dividir el cono óptico en dos problemas, uno vertical y otro horizontal.



Angulo Optico Vertical



Angulo Optico Vertical



Cono Optico

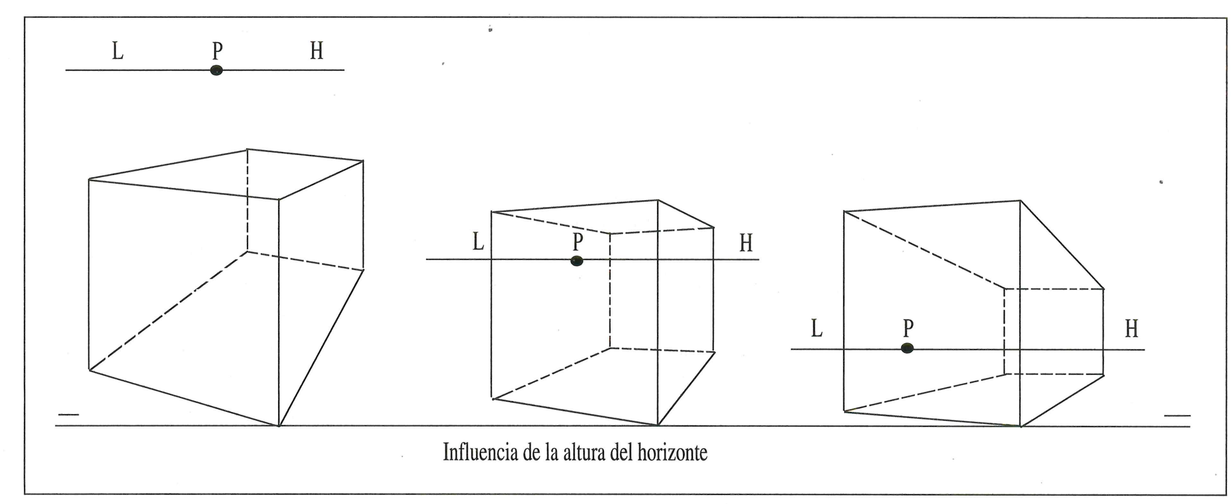


FIGURA 3

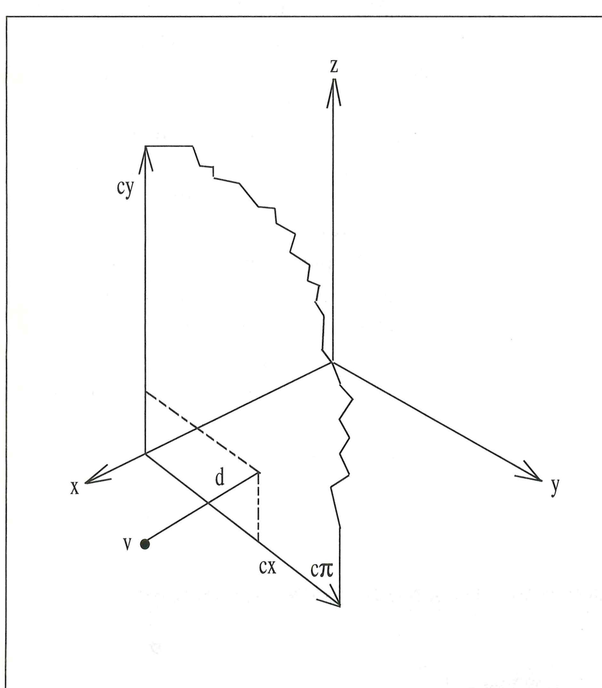


FIGURA 4
Sistema de coordenadas x-y-z y sistema de coordenadas cx-cy en el plano del cuadro.
Los planos cx-cy y z-y no tienen que ser paralelos.

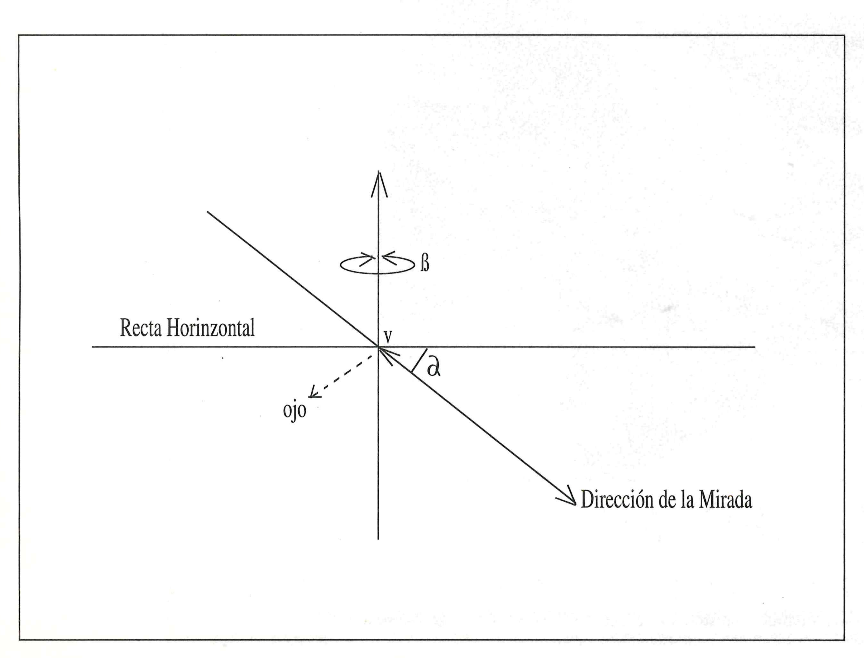


FIGURA 5
Esquema de la movilidad de nuestra cabeza una vez fijado el punto ∂

Por último, para tener definido completamente el entorno, podemos suponer que el punto V no representa a nuestros ojos sino la posición de la cabeza, teniendo así la posibilidad de definir el ángulo que giramos el cuello, tanto verticalmente (∂) como horizontalmente (ß) (para simplificar no damos la posibilidad al lector de mover la cabeza de lado, lo cual supondría un tercer ángulo). De este modo las coordenadas de V nos dan la posición del punto de mira mientras los ángulos ∂ y ß nos definen la dirección de la mirada . (Ver figura 5).

Tenemos definidas ahora las variables del entorno:

vx - coordenada x del punto de mira
vy - coordenada y del punto de mira.
vz - coordenada z del punto de mira.

d - distancia que separa al punto de mira del plano de cuadro.
∂ -ángulo vertical del «punto» de mira respecto al plano horizontal x-y
(ver figura 5).
ß -ángulo horizontal del «punto» de mira respecto al plano vertical x-z (ver figura 5 desde arriba).

Ahora el problema se reduce a tomar las coordenadas de cualquier punto M (mx, my, mz) respecto a los ejes x-y-z y transformarlas en (mcx, mcy) respecto de los ejes cx-cy. Para ello en primer lugar debemos girar el eje de coordenadas sobre z a fin de que y sea paralelo a cx, y posteriormente repetimos la operación sobre el eje y a fin de que z sea paralelo a cy. Para ello aplicamos las siguientes fórmulas:

mx' = my sin(ß) + mx cos(ß)
my' = my cos(ß) - mx sin(ß)

mz' = mx' sin(∂) + mz cos(∂)
mx''= mx' cos(∂) - mz sin(∂)

Realizamos también la misma operación con el punto de mira V, obteniendo las nuevas coordenadas (vx'', vy', vz'), que no es más que el punto V en el nuevo sistema de referencia.
Ahora, cuando el plano y'z' y cx cy son paralelos (y no por casualidad como ocurría en la figura 4) podemos resolver los problemas vertical y horizontal que sólo es hallar la intersección entre la recta que va desde (Vx'', Vy', Vz) hasta (Mx'', My', Mz) con el plano del cuadro que ahora es C-π={(x,y,z) | / x=Vx''-d (x,z) E R2)}, pero que se limita a hallar el punto donde se cortan dos rectas. Resolviendo el problema vertical, figura 6, tenemos:

$$MCy= \frac{-dz' + dVz' + x''Vz' - Vx''Vz'}{x'' - Vx''}$$

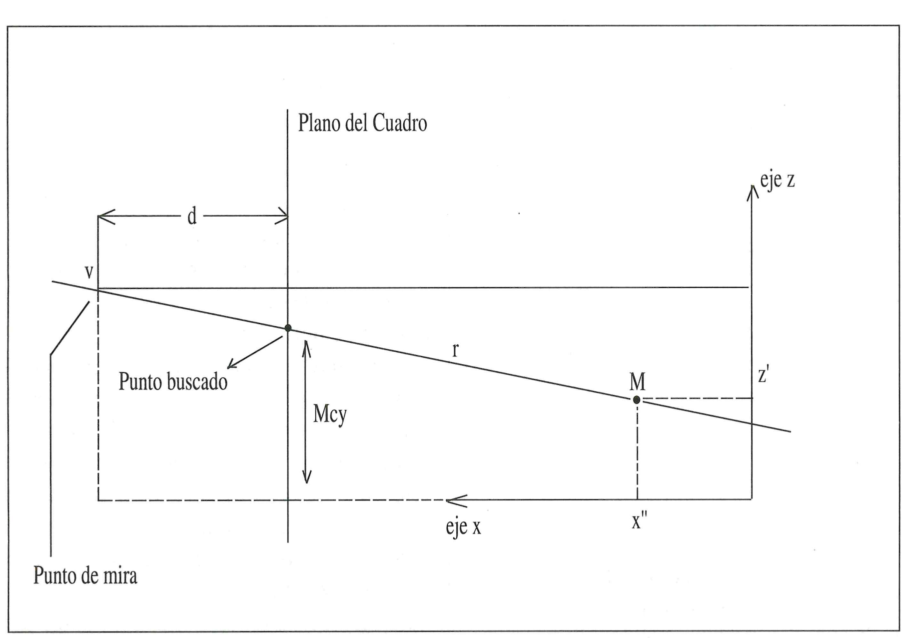


FIGURA 6
Esquema del problema vertical, hay que buscar Mcy

Resolviendo el problema horizontal, que es idéntico (cambiando la z por la y) al vertical obtenemos que:

$$MVx= \frac{-dy' + dVy' + x''Vy' - Vx''Vy'}{x'' - Vx''}.$$

Finalmente, a fin de presentar en el monitor la parte del plano del cuadro que está ante nuestros ojos hay que realizar una última transformación que consiste simplemente en sumar a todos los puntos (320-Vy', 200-Vz'). Esto es, una traslación de ejes (ver en el listado 1 la rutina dibuja_cuadrado ()).
Con estas fórmulas, y comprendiendo bien lo expuesto anteriormente, es posible representar cualquier figura definida por puntos (y por tanto rectas) en perspectiva (lo que se conoce como representación vectorial en 3D). Invitamos a los lectores que dispongan de tiempo intenten hacer funcionar el listado 1, el cual presenta en pantalla un cuadrado que está en el plano x-y, que el cruce de sus diámetros está en el punto (0,0,0) y su lado mide 100 puntos, pudiendo modificar con el teclado las variables del entorno, posición de la cabeza, dirección de la mirada y distancia del plano de cuadro.
En el programa propuesto, para modificar las variables que aparecen en pantalla (figura 8) debemos pulsar las teclas 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, (y).
Con este ejemplo se pretende simplemente

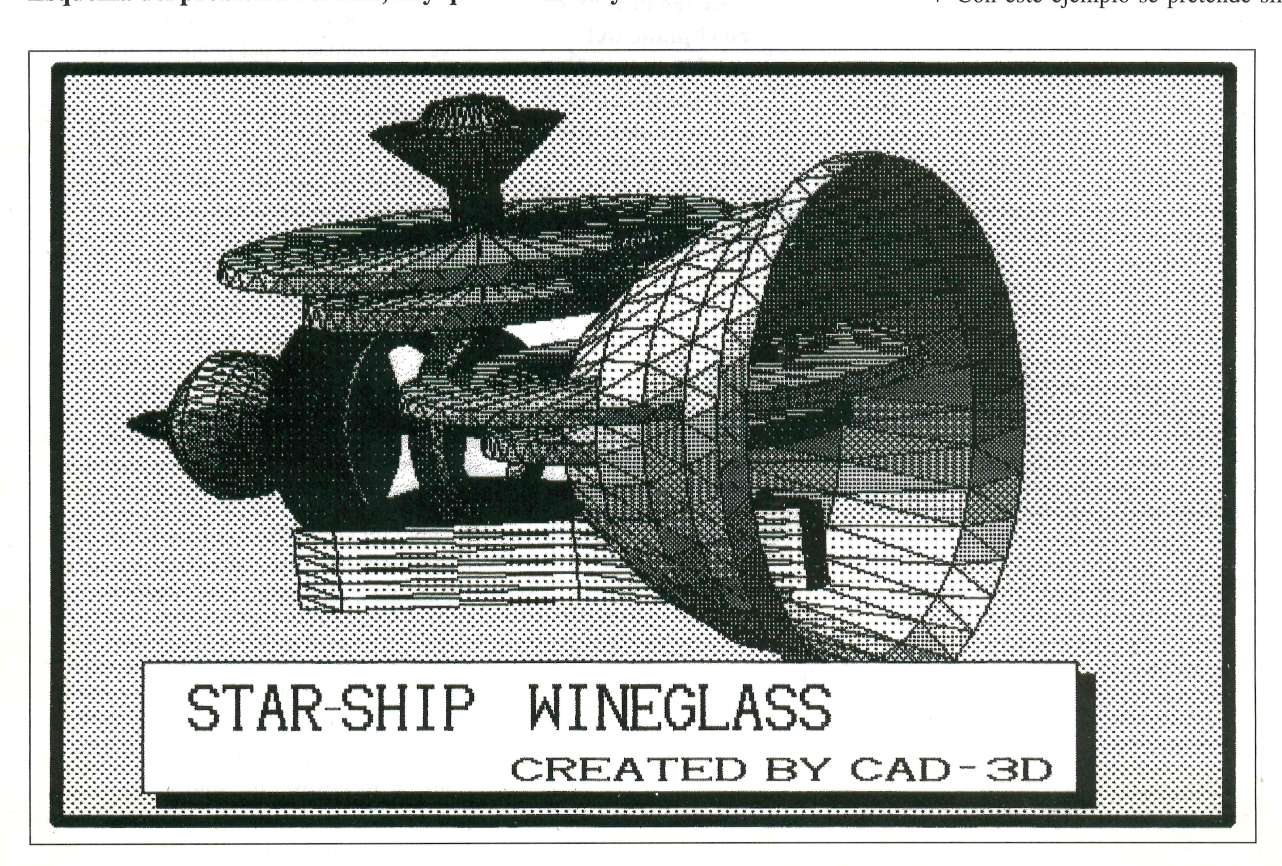


FIGURA 7 Nave realizada a partir de copas de vino con el programa CAD 3D

exponer los principios de la representación vectorial en 3D y por tanto el desarrollo de posibles programas debe ser realizado por el lector de ATARI USER.

A la teoría anteriormente expuesta se le pueden añadir muchos capítulos, que pueden ser deducidos por el lector que haya comprendido este artículo. Algunas sugerencias son la representación de volúmenes sencillos, teniendo en cuenta líneas ocultas.

Si por ejemplo deseamos representar un cubo teniendo en cuenta las líneas ocultas, deberemos tener presente que sólo podemos ver simultáneamente tres caras y por tanto para dibujar los lados del cubo sin que aparezcan los lados ocultos, deberemos dibujar sólo aquellos lados que no contengan el vértice más lejano al punto de mira.

Podemos hallar un método similar para deducir cuales son las líneas ocultas para cualquier poliedro regular o poliedros irregulares estudiados (creados aposta).

Para figuras volúmicas complejas, el buscar líneas ocultas puede suponer hasta horas de cálculo, tal y como sucede en algunos programas de diseño mal programados, como por ejemplo el AUTOCAD de PC.

Un truco, que utiliza el CAD 3D de ATARI, es dividir las figuras volumétricas en planos de distintos tamaños y ordenarlos desde el que está más lejano al punto de mira hasta el más cercano, para posteriormente dibujarlos en ese orden, de este modo los planos que estén encima irán tapando los que están debajo.

Otro posible método para dibujar figuras volumétricas teniendo en cuenta líneas ocultas consiste en ordenar los planos que las componen desde el más cercano a nosotros hasta el más lejano e irlos dibujando en la pantalla de modo que cuando parte de la pantalla está cubierta, nada puede dibujarse en dicha parte, ya que estaría en zona oculta.

Otra posible continuación en el estudio de la perspectiva, podría ser en un intento de generar imágenes en 3 dimensiones estéreas, de esas que hay que ver con gafas estereotek (un ojo azul y otro rojo). Para ello habría que crear una imagen con dos puntos de mira contenidos en un mismo plano de desvanecimiento paralelo al plano del cuadro. Un nuevo parámetro sería la separación entre ambos puntos de mira, que sería el parámetro que daría la sensación de estéreo.

Con dos puntos de mira se crearían dos imágenes, una en azul y otra en rojo, y los puntos que se superponen deberías tener un tercer color con componente azul y roja (morado). De este modo tendríamos una imagen estérea en 3 dimensiones reales.

Confiamos, con este artículo, haber desenmascarado el misterio de la generación de imágenes en tres dimensiones, y que cualquier lector pueda fácilmente utilizar sus principios. Si alguno de vosotros quisiera saber más, no hubiera entendido algo o desearía realizar alguna sugerencia o comentario, puede escribirnos a nuestra redacción, ya que como sabéis estas páginas son vuestras y podéis escribir en ellas lo que queráis.

Por A. Miguel.
Análisis matemático: Fernando Perla.

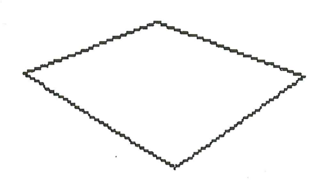


FIGURA 8 Representación de un cuadro en tres dimensiones

```
/*
Programa ejemplo didactico para la representacion de imagenes en

              3   D I M E N S I O N E S

     (C) 1991 Ibercomp SRL - CBC Press, ideado y escrito por:
                   A. Miguel & Fernando Perla

        Realizado en Laser C version 1.1, de Megamax Inc.
*/
/*Variables del GEM*/
int contrl[12];
int intin[256],  ptsin[256];
int intout[256], ptsout[256];

#include <osbind.h>
#include <gemdefs.h>
#include <obdefs.h>
#include <math.h>

/*defino las variables globales y les asigno su valor inicial.*/
int vx=800,vy=0,vz=100;
int d=654;
int alfa=0,beta=0;

/*defino las variables que han de soportar los vertices del
cuadrado,
tanto en 3 dimensiones como en dos*/
float cuadrado[4][3];
int vertice[5][2];
int vertice1[5][2];

/*Variables del GEM - Solo hacen falta en C y Assembler no en
BASIC*/
int phys_handle;
MFDB pitos;
int piniento;
MFDB*consola;

/*Defino variables que no devuelven tipo int o void.*/
extern float gradiza();

main()
{
  register int a;
  /*Inicialize la aplicacion y abre una estacion virtual de
trabajo*/
  appl_init();
  piniento=open_vwork(&pitos);
  inicializa_sistema();
  dibuja_cuadrado();
  do {
    a=Bconin(2);
    switch (a) {
      case '1':
        if (vx<32000) {
          vx+=5;
          locate(2,6);
          escribe("%5d",vx);
        }
        break;
      case '2':
        if (vx>-32000) {
          vx-=5;
          locate(2,6);
          escribe("%5d",vx);
        }
        break;
      case '3':
        if (vy<32000) {
          vy+=5;
          locate(3,6);
          escribe("%5d",vy);
        }
        break;
      case '4':
        if (vy>-32000) {
          vy-=5;
          locate(3,6);
          escribe("%5d",vy);
        }
        break;
      case '5':
        if (vz<32000) {
          vz+=5;
          locate(4,6);
          escribe("%5d",vz);
        }
        break;
      case '6':
        if (vz>-32000) {
          vz-=5;
          locate(4,6);
          escribe("%5d",vz);
        }
        break;
      case '7':
        if (alfa<359) {
          alfa++;
        }
        else {
          alfa=0;
        }
        locate(2,27);
        escribe("%3d",alfa);
        break;
      case '8':
        if (alfa>0) {
          alfa--;
        }
        else {
          alfa=359;
        }
        locate(2,27);
        escribe("%3d",alfa);
        break;
      case '9':
        if (beta<359) {
          beta++;
        }
        else {
          beta=0;
        }
        locate(3,27);
        escribe("%3d",beta);
        break;
      case '0':
        if (beta>0) {
          beta--;
        }
        else {
          beta=359;
        }
        locate(3,27);
        escribe("%3d",beta);
        break;
      case '(':
        if (d<32000) {
          d+=5;
          locate(2,52);
          escribe("%5d",d);
        }
        break;
      case ')':
        if (d>100) {
          d-=5;
          locate(2,52);
          escribe("%5d",d);
        }
        break;
    }
    dibuja_cuadrado();
  } while(a!=27);
  v_clsvwk(piniento);
  appl_exit();
}

inicializa_sistema()
{
  int inutil,v[4];

  v[0]=0;
  v[1]=80;
  v[2]=639;
  v[3]=399;

  /*Primero busca el canal de salida y pone modo REPLACE para las
lineas.*/
  phys_handle=graf_handle(&inutil,&inutil,&inutil,&inutil);
  vswr_mode(phys_handle,1);
  vs_clip(phys_handle,1,v);

  cls();
  locate(0,6);
  escribe("Programa Didactico Educativo Generados de Imagenes en 3
Dimensiones\n\r");
  locate(2,1);
  escribe("VX - %5d",vx);
  locate(3,1);
  escribe("VY - %5d",vy);
```

```
  locate(4,1);
  escribe("VZ - %5d",vz);

  locate(2,20);
  escribe("ALFA - %3d°",alfa);
  locate(3,20);
  escribe("BETA - %3d°",alfa);

  locate(2,40);
  escribe("Distancia - %5d",d);
}

cls()
{
  /* Borra la pantalla. */
  Bconout(2,27);
  Bconout(2,'E');
}

locate(a,b)
  register int a,b;
{
  /* Coloca el cursor en a,b */
  Bconout(2,27);
  Bconout(2,'Y');
  Bconout(2,32+a);
  Bconout(2,32+b);
}

escribe(arg1,arg2)
  char *arg1;
  int arg2;
{
  char a[100];
  register char *b;
  _sprintf(a,arg1,&arg2);
  b=a;
  while (*b) {
    Bconout(2,*b);
    b++;
  }
}

dibuja_cuadrado()
{
  static int p=0;
  float vx1=vx,vy1=vy,vz1=vz;
  register int x;

  /*asignamos al cuadrado*/
  cuadrado[0][0]=-50.0;
  cuadrado[0][1]=50.0;
  cuadrado[0][2]=0.0;

  cuadrado[1][0]=50.0;
  cuadrado[1][1]=50.0;
  cuadrado[1][2]=0.0;

  cuadrado[2][0]=50.0;
  cuadrado[2][1]=-50.0;
  cuadrado[2][2]=0.0;

  cuadrado[3][0]=-50.0;
  cuadrado[3][1]=-50.0;
  cuadrado[3][2]=0.0;

  /*Transformamos el cuadrado y el punto de mira*/
  transforma(&cuadrado[0][0],&cuadrado[0][1],&cuadrado[0][2]);
  transforma(&cuadrado[1][0],&cuadrado[1][1],&cuadrado[1][2]);
  transforma(&cuadrado[2][0],&cuadrado[2][1],&cuadrado[2][2]);
  transforma(&cuadrado[3][0],&cuadrado[3][1],&cuadrado[3][2]);
  transforma(&vx1,&vy1,&vz1);

  /*Hallamos los nuevos vertices en 2 D*/
  for (x=0;x<4;x++) {

vertice[x][0]=bidimensiona(vx1,vy1,cuadrado[x][0],cuadrado[x][1])+3
20-(int)vy1;

vertice[x][1]=399-(bidimensiona(vx1,vz1,cuadrado[x][0],cuadrado[x][
2])+200-(int)vz1);
  }
  vertice[4][0]=vertice[0][0];
  vertice[4][1]=vertice[0][1];

  if (p) {
    vsl_color(phys_handle,0);
    Vsync();
    v_pline(phys_handle,5,vertice1);
  }
  else {
    p=1;
  }
  vsl_color(phys_handle,1);
  v_pline(phys_handle,5,vertice);

  for (x=0;x<5;x++) {
    vertice1[x][0]=vertice[x][0];
    vertice1[x][1]=vertice[x][1];
  }
}

transforma(mx,my,mz)
  float *mx,*my,*mz;
{
  float x,y,z;
  float a_alfa,a_beta;
  a_alfa=gradiza(alfa);
  a_beta=gradiza(beta);

  x=*my*sin(a_beta)+*mx*cos(a_beta);
  y=*my*cos(a_beta)-*mx*sin(a_beta);

  z=x*sin(a_alfa)+*mz*cos(a_alfa);
  *mx=x*cos(a_alfa)-*mz*sin(a_alfa);
  *my=y;
  *mz=z;
}

bidimensiona(vx2,vy2,x,y)
  float vx2,vy2,x,y;
{
  register int
n=(int)(((float)d*y-(float)d*vy2-x*vy2+vx2*vy2)/(-x+vx2));
  return n;
}

float gradiza(n)
  int n;
{
  return (float)n*3.141549/180.0;
}

int open_vwork(form)
  register MFDB*form;
{
  register int x;
  int  work_in[11];
  int handle, d;
  int work_out[57];

  /*
    Inicializa las variables de la estacion de trabajo virtual.
  */
  for(x=0; x<10; x++)
    work_in[x] = 1;

  work_in[10] = 2;
  handle = graf_handle(&d, &d, &d, &d);
  v_opnvwk(work_in, &handle, work_out);
  form -> fd_addr = Logbase();
  form -> fd_w= work_out[0] + 1;
  form -> fd_h= work_out[1] + 1;
  form -> fd_wdwidth= form -> fd_w /16;
  form -> fd_stand= 0;

  switch(work_out[13]) {
    case 16: form -> fd_nplanes = 4; break;
    case 08: form -> fd_nplanes = 3; break;
    case 04: form -> fd_nplanes = 2; break;
    default: form -> fd_nplanes = 1; break;
  }
  consola=form;
  return handle;
}
```

LISTADO 1 Programa que representa un universo con un cuadrado, en el cual podemos pasearnos

INTERRUPCIONES EN EL ATARI ST/TT

En el pasado número de ATARI USER intentábamos explicaros básicamente lo que son las interrupciones y os comentábamos que en este siguiente capítulo comenzaríamos con un sencillo ejemplo de cómo enganchar una interrupción al VBL por el medio legal, así pues, vamos a ello.

En el pasado número de ATARI USER intentábamos entender básicamente lo que son las interrupciones y os comentábamos que en este siguiente capítulo comenzaríamos con con un sencillo ejemplo de cómo enganchar una interrupción al VBL por el medio legal, así pues, continuamos.
Este primer ejemplo está escrito en lenguaje C, por ser la única interrupción que puede usarse bajo el lenguaje C sin meter una sola línea de ensamblador.
Obviamente desde BASIC, PASCAL, ... el uso de las interrupciones está totalmente descartado.
(ver ejemplo 1)

Imagino que todos los que hayais probado este programita escrito en lenguaje C (en concreto para el LASER C de MEGAMAX) sabréis de sobra lo que es una interrupción, o sea que el concepto de interrupción está claro.
En lenguaje C puro no se puede hacer mucho más con interrupciones. Hay que tener en cuenta que ninguna interrupción debe acceder al sistema operativo, BIOS, XBIOS, GEMDOS, GEMVDI, ... tan sólo A Line puede ser llamado. En caso de llamar a dichas rutinas, el cuelgue más o menos temprano está asegurado.
Para los que programan directamente en código máquina, les hacemos saber que las rutinas enganchadas al vbl pueden modificar todos los registros a excepción de los A7. Así mismo la rutina debe acabar con un rts.
ejemplo:

```
; Esta función es equivalente a la contador() del
; ejemplo en C anterior.

contador
      addq.w   #1,p
      rts

p: dc.w   0
```

El siguiente ejemplo nos servirá para demostrar cómo podemos utilizar el vbl directamente para nuestros fines.
Normalmente la utilidad de esta rutina es para evitar los parpadeos de pantalla, o sea para poder realizar Scrolles sin que parpadee la pantalla, el siguiente ejemplo en código máquina lo demuestra.

Para todos los lectores de ATARI USER que tengan nociones de código máquina y que sea la primera vez que ven un proceso de interrupción, les hacemos saber que siempre que se salta a una interrupción para volver hay que hacerlo con RTE en vez de RTS.
RTE en un 68000 es equivalente a hacer un move.w (A7)+, SR seguido de un TRS. Decimos más o menos equivalente porque en un 68030 (TT) RTE hace algo más, primero saca de la pila el registro ofset, a continuación el SR y al final hace un RTS. ¿Por qué?
Muy sencillo, para permitir a las interrupciones poder dejar los programas en las mismas condiciones en las que estaban, es imprescindible guardar en la pila todos los registros que vayan a ser utilizados por esta y restaurarlos antes de realizar el RTE. Hay un registro que es modificado queramos o no queramos, este es el SR, que contiene entre otras cosas un bit que nos indica si el ordenador está en supervisor o no, este bit es puesto a uno al entrar en la interrupción, con lo que el ordenador guarda el SR antes de entrar en la interrupción. Otro registro que siempre es modificado, como es obvio es el PC, que es guardado en la pila después del SR. (De aquí podemos deducir que RTS es lo mismo que move.l (A7)+,PC, aunque esta última nomenclatura no se utiliza).
Fuera de este detalle RTE no presenta ningún problema, excepto que debe ser utilizado en modo supervisor.
Es interesante saber que es posible aprovecharse de RTS y RTE para obligar al nuevo_vbl que llame al antiguo, esto se consigue:

```
nuevo_vbl
        movem.l D0/A0-A1,-(A7)
        .
        . ;Proceso que utilice D0, A0 y A1.
        .
        movem.l (A7)+,D0/A0-A1
        move.l viejo_vbl,-(A7)
        rts
        .
        .
        .
```

Este RTS utilizado aquí no volverá al programa que ha sido interrumpido, sino que irá al viejo_vbl.
Aprovecho para comentar a los programadores en código máquina que la utilización de cosas como:

```
chapuza:
        move.l viejo_vbl,salto+2
salto:
        jmp      0
```

funcionan en un 68000, pero no tienen porque funcionar en el 68030, es más todos los programas que utilizan técnicas de autoprogramación tenderán a colgarse, debido a que el 68030 tiene una memoria de datos y una memoria de programas «independiente» (mismas direcciones, pero distintos valores, dicho de otro modo, el TT hará un jmp 0 y no un jmp viejo_vbl como cabe esperar).
La interrupción vbl la doy por explicada, y resumo diciendo que es ejecutada cada vez que el ordenador termina de enviar una pantalla al monitor, de este modo podemos estar seguro que en ese momento podemos cambiar la pantalla y/o dibujar cualquier cosa sin que se note ningún tipo de parpadeo. Propongo que los lectores que crean haber comprendido las ideas básicas se esfuercen en realizar un programa que saque en modo monocromo negro, gris y blanco sin utilizar tramas.

En el siguiente número seguiremos explicando detalladamente las interrupciones de MFP (Multi Funcion Peripheral), un periférico multifunción.

```
/*
  Ejemplo de enganchar interrupciones al vbl.
*/

/* Definimos las variables nvbls.w y vblqueue. */

#define nvbls    *(int *)0x454
#define vblqueue (*(long **)0x456)

/*  Incluimos  el osbind para poder acceder a  Super,  Bconstat  y
    Bconin.*/

#include <osbind.h>

/* preDefinimos las funciones a enganchar como externas.*/

extern contador(),alt_help();

/*  Definimos  p  como  variable  global,   la  utilizaremos  como
    contador.*/

int p;

main()
{
  enganchar(contador);   /*Enganchamos la rutina contador*/
  enganchar(alt_help);   /*y la rutina alt_help*/

  /* Realizamos un bucle hasta que se pulse una tecla
     (hacerlo con insistencia), en el cual imprimimos la
     variable p que no cambia de valor.
  */
  do {
    printf(">> %05d\n\r",p);
  } while (!inkey());
  desenganchar(alt_help); /*Desenganchamos las rutinas*/
  desenganchar(contador);
}


/*  A  continuación la rutina de enganchar rutinas al  vbl,  para
    ello hay que pasarle como parametro la direccion de la rutina
    a enganchar.
*/
enganchar(puntero)
  long puntero;
{

/* Definimos  x  y encontrado,  y entramos  en  modo  supervisor
   función  Super(0L),  algo necesario para acceder a  cualquier
   variable del sistema.
*/
register int x,encontrado=0;
long stack=Super(0L);

/* Bucle que busca si hay un vblqueue vacio,  esto es a cero,  en
   caso  afirmativo  mete  en el la dirección  de  la  rutina  a
   enganchar.
*/
for (x=0;x<nvbls;x++) {
  if (!vblqueue[x]) {
    encontrado=1;
    break;
  }
}
if (encontrado) {
  vblqueue[x]=puntero;
}

  /* Si no se ha encontrado ningun hueco, no puede enganchar, para
     ello  habría  que escribir un programa que nos  aumentase  el
     tamaño, tal y como se explica arriba.
  */

  else {
    printf ("No hay espacio para enganchar otra rutina\n\r");
  }
  /* Volvemos  a modo usuario permanecer siempre el tiempo  minimo
     en supervisor*/.

  Super(stack);
}

/*  Similar a la función de arriba, solamente que desengancha una
    función ya enganchada.
*/

desenganchar(puntero)
  long puntero;
{
  register int x,encontrado=0;
  long stack=Super(0L);
  for (x=0;x<nvbls;x++) {
    if (vblqueue[x]==puntero) {

     encontrado=1;
     break;
   }
 }
 if (encontrado) {
   vblqueue[x]=0L;
 }
 else {
   printf ("Dicha rutina no esta enganchada.\n\r");
 }
 Super(stack);
}

/*  Función que incremente la variable p. */
contador()
{
  p++;
}

/*   Función que pone a 0 la variable p cada vez que dmpflg  vale
     0, esto es cada vez que se pulsa ALT HELP, a continuación hace
     -1  a  dmpflg  para  que  no se  imprima  el  invento  por  la
     impresora.
*/

alt_help()
{
  #define dmpflg (*(int *)0x4ee)
  if (!dmpflg) {
    /* Aqui sabemos que ALT HELP ha sido pulsado. */
    p=0;
    dmpflg=-1;
  }
}


/*
   Simula a la sentencia inkey del Basic, devuelve el codigo ASCII
   de la tecla que ha pulsado o 0 sino se ha pulsado ninguna.
*/
inkey()
{
  if (Bconstat(2)) {
    return Bconin(2)&255;
  }
  return 0;
}
```

```
; Esta  función  es  equivalente a  la  contador()  del
; ejemplo en C de arriba.

contador:
     addq.w    #1,p
     rts

p:   dc.w      0
 ;
 ;Ejemplo en ensamblador de como utilizar el vbl
 ;
           ;pido al sistema operativo la dirección de la pantalla
           ;fisica.
           move.w  #2,-(a7)
           trap    #14
           addq.l  #2,A7
           addi.l  #200*80,D0      ;Para que se escrolee por el
                                   ;centro, se puede omitir.
           move.l  D0,direccion

           ;Pasa a modo supervisor.
           clr.l   -(A7)
           move.w  #$20,-(A7)
           trap    #1
           addq.l  #6,D7
           move.l  D0,stack

           ;A continuación se llena la pantalla de "A" dentro de
           ;lo posible. (Se puede mejorar bastante)
           move.w  #23,D0
 bucle1:
           move.w  D0,-(A7)
           move.w  #78,D0     ;poner 38 en vez de 78 en baja
                              ;resolución
 bucle2:
           move.w  D0,-(A7)
           move.w  #'A',-(A7)
           move.w  #2,-(A7)
           trap    #1
           addq.l  #4,A7
           move.w  (A7)+,D0
           dbf     D0,bucle2
           move.w  #13,-(A7)
           move.w  #2,-(A7)
           trap    #1
           addq.l  #4.A7
           move.w  #10,-(A7)
           move.w  #2,-(A7)
           trap    #1
           addq.l  #4,A7
           move.w  (A7)+,D0
           dbf     D0,bucle1

           ;Guarda el antiguo vbl.
           move.l  $70,viejo_vbl
           ;Mete el nuevo vbl.
           move.l  #nuevo_vbl,$70

 bucle3:   ;Se espera a que se pulse una tecla.
           move.w  #1,-(A7)
           trap    #1
           addq.l  #2,A7

           ;Se restaura el vbl.
           move.l  viejo_vbl,$70

           ;Se vuelve al modo usuario.
           move.l  stack,-(A7)
           move.w  #$20,-(A7)
           trap    #1
           addq.l  #6,D7

           ;Se indica al sistema operativo que el programa se
           ;ha acabado. -> Salir del programa.
           clr.l           -(A7)
           trap    #1

           ;Este es el nuevo vbl, sustituye al antiguo, con lo que
           ;cualquier cosa que incluya cambio de monitor, unidad
           ;de disco encendida, rutinas enganchadas al vbl, ...,
           ;seran ignoradas, nuesto vbl sustituye totalmente al
           ;anterior. Lo unico que hace es mover un bloque de
           ;memoria por dentro de la pantalla, y nada más.
nuevo_vbl:
           ;En primer lugar hay que guardar los registros
           ;utilizados en la pila, ya que el valor de estos
           ;pertenecen al programa interrumpido y no al nuevo_vbl.
           movem.l D1/A0-A1,-(A7)
           ;A continuación cogemos la dirección de la pantalla, y
           ;movemos unos cuantos bytes dentro de ella.
           move.l  direccion,A0
           move.l  A0,A1
           adda.l  #80,A0
           move.w  #160-1,D1

bucle4:
           move.l  (A0)+,(A1)+
           move.l  (A0)+,(A1)+
           move.l  (A0)+,(A1)+
           move.l  (A0)+,(A1)+
           dbf     D1,bucle4
           move.l  direccion,A0
           move.w  #19,D1
bucle5:
           move.l  (A0)+,(A1)+
           move.l  (A0)+,(A1)+
           dbf     D1,bucle5
           ;Restauramos los registros utilizados y volvemos
           ;inmediatamente al programa interrumpido.
           movem.l (A7)+,D1/A0-A1
           rte

direccion:
           dc.l        0
stack:
           dc.l        0
viejo_vbl:
           dc.l        0

nuevo_vbl
        movem.l D0/A0-A1,-(A7)
           .
           .    ;Proceso que utilice D0, A0 y A1.
           .
        movem.l (A7)+,D0/A0-A1
        move.l  viejo_vbl,-(A7)
        rts
           .
           .
           .

     chapuza:
           move.l viejo_vbl,salto+2

   salto:
           jmp     0
```

Juegos

PREVIEWS

• COLORS
• Infogrames
En la línea de rompecabezas-arcade, Colors es el digno sucesor de Tetris. Se juega contra el ordenador o a dos, el objetivo del juego es el de conquistar el máximo de terreno. Este último está constituido por una multitud de pequeñas losetas de colores para aumentar la superficie hay que conjugar tiempo y colores.

• OUTZONE
• Lankhor
Antes de presentarnos Vroom, Lankhor edita otro juego de tipo distinto, Outzone. Este shoot´em up pone a prueba tu capacidad de reflexión y claro, tu puntería. Compuesto por 28 misiones, Outzone se beneficia de una suntuosa realización con 50 imágenes por segundo en el ST. Se prevee su salida para el 92.

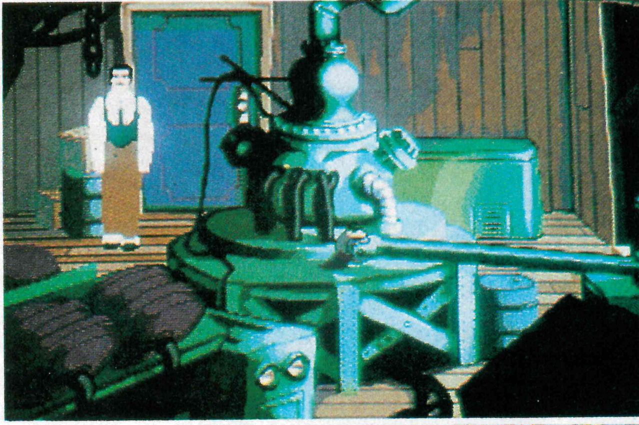
• BOOLY
• Loriciel
Juego de reflexión por excelencia, Booly hace esencialmente llamada a tus dotes de observación y memoria, sobre todo si juegas sin ayuda.
Se trata de convertir todos los elementos de un nivel a un mismo símbolo (siguiendo el principio de «binariedad»), estos elementos están liados entre sí por uniones bivalentes,
Muy fácil al principio va siguiendo un orden creciente de complejidad, no es extraño teniendo en cuenta que posee más de 1.000 niveles. También para el 92.

• CROISIERE POUR UN CADAVRE
(Crucero para un cadáver)
• Delphine Software
Esta vez sí es la buena. Saldrá para este mes, el equipo de Delphine a cambiado totalmente el juego, para al final, llegar prácticamente a la perfección en el dominio de los juegos de aventura. Los gráficos son superiores, el escenario perfecto y la animación de los personajes extraordinaria. Sale en Francia ya.

• WIZKID
• Ocean
Previsto que salga en verano para Amiga, Wizkid no verá la luz en el 91 para ST. El programa que se presenta como la continuación de Wizball, es fundamentalmente diferente, y sobre todo enloquecedor y delirante, en él se mezcla arcade y reflexión. Además, la realización se prevee superior.

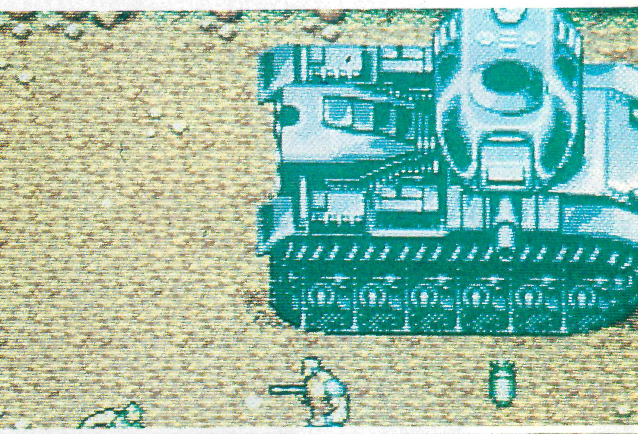
• MERCS
• US Gold
Se trata de un juego del estilo «comando», se puede jugar solo o a dos. El objetivo es el de liberar al presidente de los Estados Unidos, caído en manos de la guerrilla. Gráficos simpáticos, explosiones por todas partes, combates incesantes... Mercs complacerá a todos los "Rambo" en vena.

• SHADOW SORCERER
• SSI
Los fans de Donjons & Dragons podrán disfrutar con Shadow Sorcerer desde este mes. Con un escenario inspirado en el mundo de Krynn, el juego propone un





CROISIERE POUR UN CADAVRE



MERCS



MERCS

17.900* Ptas. + IVA

* NO INCLUYE PERIFERICOS NI JUEGOS

ATARI LYNX

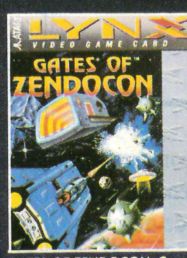
LYNX

EL PRIMER SISTEMA DE ENTRETENIMIENTO

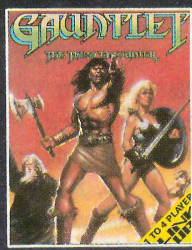
ACCION



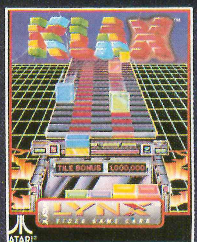
ELECTROCOP. Eres un Poli del siglo XXI, mitad hombre mitad robot. Te han encargado liberar a la hija del presidente encerrada en un complejo con varios niveles.



GATES OF ZENDOCON. Cargado con múltiples y devastadoras armas, conduces tu nave a través de 99 asombrosos mundos, rechazando las incesantes oleadas de alienígenas.



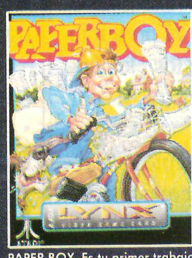
GAUNTLET. Combate guerreros a través de 40 niveles para 4 aventureros que descubrirán pócimas y otros tesoros.



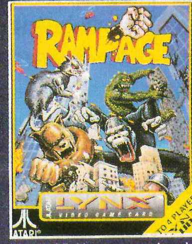
KLAX. Para ser un maestro de KLAX preciso disponer de una buena dosis de destreza, lógica y autocontrol. ¡Atrévete con el desafío!



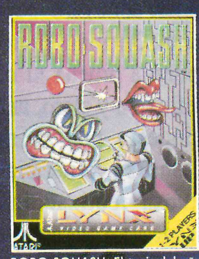
s. PAC MAN. ¡Gobble, Gobble, slub! Miss Pac Man, la diosa de los laberintos, ha regresado. Engulle las píldoras y trágate a los fantasmas.



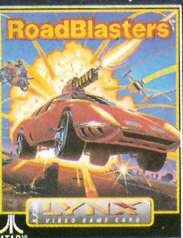
PAPER BOY. Es tu primer trabajo: repartidor del Daily Sun. Se trata de dar buena impresión, pero es difícil distribuir el periódico evitando los obstáculos.



RAMPAGE. No es necesario volar el edificio para destruirlo. Te basta con un puñetazo. Y si los soldados vienen a hacerte cosquillas en los dedos de los pies, cómetelos.



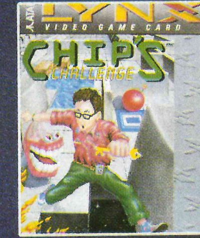
ROBO SQUASH. El tenis del año 2000 se juega en salas. Tu eliges, o tu adversario es el Lynx u otro jugador conectado gracias al Comlynx.



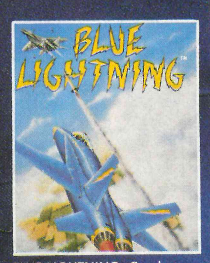
ROAD BLASTERS. En un mundo devastado por un conflicto atómico, burlas a los Mad Max, pisando el acelerador a fondo, durante un rally de 50 etapas.



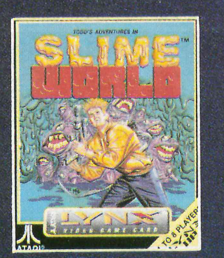
RYGAR. Eres el elegido Rygar el guerrero. Debes liberar la tierra de las hordas del mal que lo han invadido. Para ello tendrás que atravesar 23 paisajes.



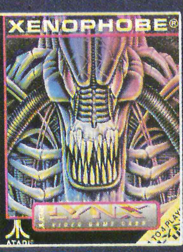
CHIP'S CHALLENGE. Sólo tu astucia y tu destreza te permitirán abrirte camino y evitar las trampas y los obstáculos de los 144 niveles de este juego.



BLUE LIGHTNING. Con los controles a tope, pilotas tu F15 Eagle a ras de tierra sobrevolando paisajes tridimensionales escarpados, destruyendo los tanques y radares enemigos.



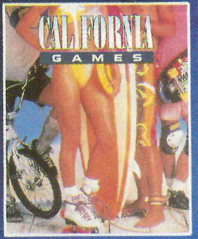
SLIME WORLD. Te vas a aventurar en un mundo pegajoso, esponjoso, verdoso, inquietante. No lo dudes, gracias al Comlynx, pueden acompañarte hasta 7 amigos en esta expedición.



XENOPHOBE. Tendrás que cometer los Xenos en más de una veintena de estaciones espaciales. Un consejo ¡no te aventures solo! Si quieres podrán acompañarte hasta tres amigos.



ZARLOR MERCENARY. Si se mueve dispara hacia arriba, si se queda quieto, no lo dudes, dispárale también. Y, gracias al Comlynx, tres amigos te pueden ayudar en tu misión.



CALIFORNIA GAMES. Todo el espíritu deportivo de California en cuatro juegos: Surfing, BMX, Footbag y Skateboarding.

3.900 Ptas + IVA

PERIFERICOS LYNX

- ADAPTADOR PARA RED DE 220 V.
- CABLE COMLYNX PARA INTERCONECTAR VARIOS LYNX.
- ADAPTADOR PARA COCHE.
- VISERA SOLAR PARA PANTALLA.
- MALETIN PARA LYNX Y ACCESORIOS.
- CARTUCHERA LYNX.



NX™
EN EL MUNDO
PORTATIL A TODO COLOR
PARA LLEVAR
Con el buen tiempo, seguro que no te apetece quedarte en casa. Sal, disfruta del sol, queda con tus amigos... y no olvides llevarte tu LYNX.
LYNX es un "todo terreno" con el realismo que estás acostumbrado a ver en las mejores máquinas de juegos:
• Pantalla de cristal liquido retroiluminada a todo color.
• Microprocesador gráfico de 16 bits, que permite la representación de mundos tridimensionales.
• Cuatro canales de sonido que dan una impresionante sensación de realismo.
• Joypad de ocho direcciones para un mayor control.
• Paleta de 4.096 colores que aporta gran calidad de imagen.
Y además...
• Puedes conectar hasta ocho LYNX para compartir la acción con tus amigos.
• Utilizarlo con auriculares (opcional).
• Alimentarlo con pilas o con adaptador para encendedor de coche o red.
• Juegos en forma de tarjeta con capacidad de hasta 8 Mb.
LYNX
ATARI
ON
OFF
OPTION 1
RESTART
FLIP
OPTION 2
B
A
LLEVATE TU LYNX,
NO HAY OTRO IGUAL.
LYNX™
LA NUEVA GENERACION.
VISITANOS EN
INFORMAT
PALACIO Nº 1
NIVEL 1 STAND 555
ATARI
ALTA TECNOLOGIA
AL MEJOR PRECIO.
ORDENADORES ATARI, S.A.: Apartado 195 - ALCOBENDAS. 28100 MADRID. TLF. (91) 653 50 11 / 663 83 36
• NORTE (947) 21 20 78 • LEVANTE (96) 362 38 61
• ANDALUCIA (95) 428 19 67 • CANARIAS (928) 36 90 81
• SOFTWARE CENTER, S.A.: CATALUÑA (93) 424 17 03 • Y en todos los centros de El Corte Inglés

SHADOW SORCERER





equipo de cuatro personas y una buenísima realización con vistas en 3 dimensiones isométricas.

• GUARDIANS
• **Loriciels**
Dentro del dominio de los juegos de reflexión, Guardians se encuentra en la cresta. El objetivo del juego es simple, evitar que uno dos o tres (en función de la dificultad del juego) triffids (unas bolas) salgan de una pieza cúbica donde unas de las seis caras está abierta, con ayuda de ladrillos. Estos ladrillos, elementos esenciales del muro a construir, se rigen por diversas reglas, incompatibilidad a nivel de color, agentes específicos, número definido para construir una zona fija...

• PEGASUS
• **Gremlin**
Puro juego de acción, Pegasus es un shootem up en el cual guiarás a Pegasus el caballo alado para enfrentarte a una horda de enemigos.
Si en principio el juego es bastante clásico, en revancha lo que si es menos, es la belleza de los gráficos y sobre todo unos fondos increíbles para un juego de este tipo.
Añadir un scroll diferencial impecable y una segunda fase del juego a lo Beast, donde el héroe cabalga por numerosas pantallas y vosotros obtenéis de nuevo un programa bastante excepcional.
Previsto para el 92.

PREVIEWS • DRO SOFT • PREVIEWS

• BUDOKAN
• **Autor: Electronic Arts**
• **Sistema: PC**
Este juego ofrece la posibilidad de aprender cuatro de las artes marciales clásicas: Karate, Kendo, Nunchaku y Bo. En ellas se combina el poder y el arte en un peligroso combate de cinturones negros. El objetivo del juego es llegar a dominar estas artes marciales. Habrá que luchar contra doce luchadores, cada uno con su propia técnica, para encontrar el camino del guerrero y llegar así a ser llamado Maestro. En caso de duda, el Sensei Tobiko estará disponible para que se le hagan las consultas necesarias. Budokan, basado en las auténticas artes marciales de Japón y Okinawa, utiliza las armas empleadas por los guerreros Samurais.
Característica: cada estilo ofrece hasta veinticinco movimientos diferentes y posibilidad de entrenar contra el ordenador o contra cualquier otro jugador, con auténticos sonidos digitalizados.

• DAS BOOT
• **Autor: Mindscape**
• **Sistema: PC**
Invierno de 1941. Nos encontra-



PEGASUS

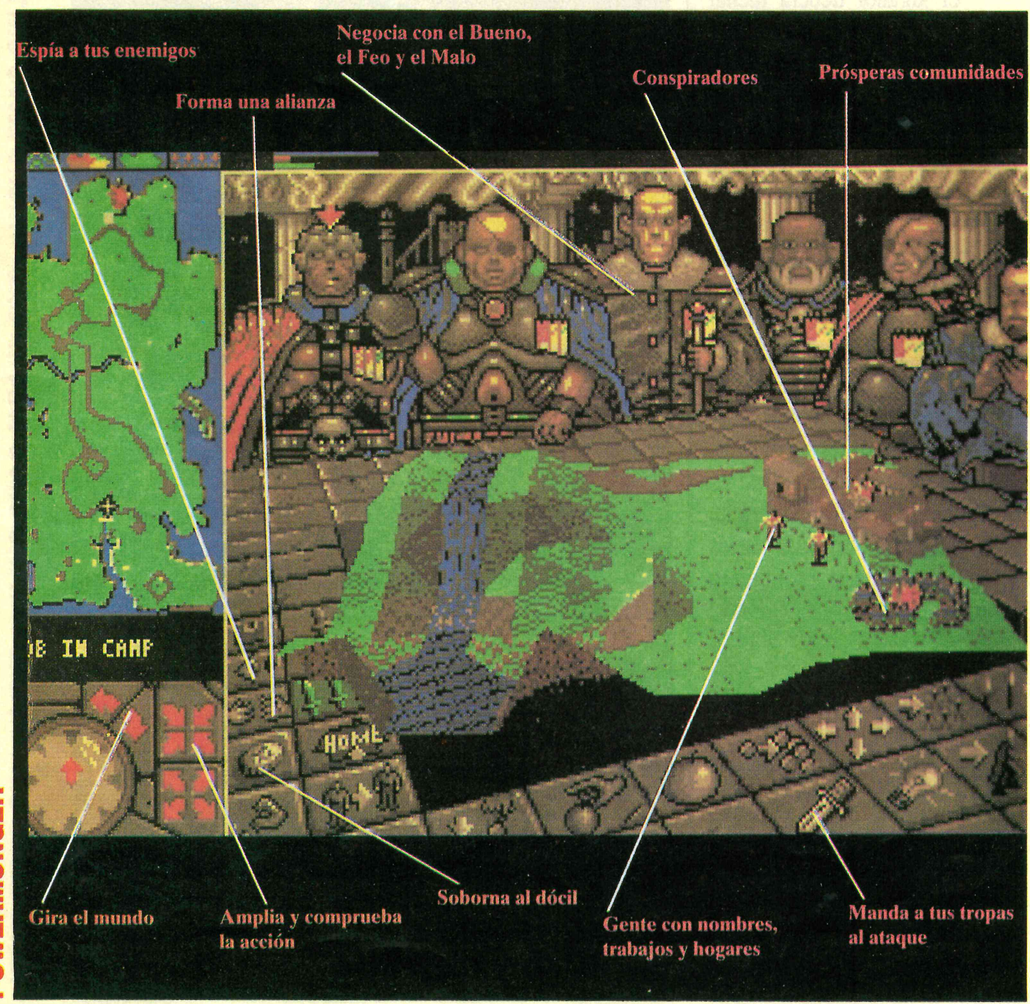
PEGASUS



mos en las heladas aguas del norte del Atlántico a 200 pies de profundidad y las cargas golpean la cubierta del submarino como si fueran puños de hierro. Ahora el jugador está al mando del submarino, uno de los más completos que existen. La misión será utilizar los torpedos para enviar al fondo a las fuerzas aliadas. Tendrá que hostigar los blancos de superficie y de tierra, y defenderse de las naves enemigas. Cuando la presión sea demasiado fuerte y los destructores enemigos estén cercando el submarino, la única solución es sumergirse y desaparecer... ¡si es que existe alguna posibilidad de hacerlo!
Característica: El submarino cuenta con una gran cantidad de dispositivos de tecnología altamente avanzada: hidrófono, máquina de códigos Enigma, cuatro tipos distintos de torpedos (de contacto, magnéticos, de bucle y acústicos). El realismo del juego es histórico. Ofrece muchas misiones entre las que elegir, tres niveles distintos de dificultad y estupendos gráficos en VGA de 256 colores con múltiples vistas internas y externas en un completo mundo tridimensional.

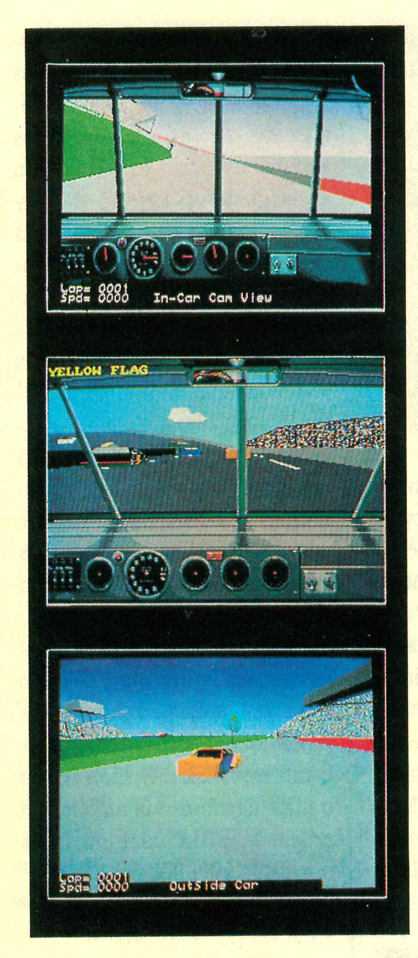
• CENTURION
• **Autor: Electronics Arts**
• **Sistema: PC**
Estamos en el año 275 a. C. Mediante la conquista y con mucha diplomacia, la República Romana ha conseguido el control de los reinados de la Península Italiana. Este es el momento en que los herederos de Rómulo y Remo entran en escena. Es el comienzo de un Imperio que llegó desde Europa hasta Asia Menor y Africa del Norte. Tú eres un ambicioso oficial y tu deber es defender Roma. Tu deseo es convertirte en César. Tu destino, gobernar el mundo.
Clave: control directo de tus tropas. Las batallas pueden desarrollarse en tierra o mar. Tú eliges la estrategia a seguir y diriges la acción. También dispones de varios niveles de juego, desde principiantes hasta avanzado.

• POWERMONGER
• **Autor: Electronic Arts**
• **Sistema: Atari**
Tu reinado de Miremer fue destruido por un devastador y violento levantamiento. Pocos de tus hombres quedaron con vida y tus tierras desaparecieron por completo. Has estado navegando por los mares durante semanas buscando un nuevo hogar para los seguidores que te quedan (y un nuevo reino para ti). Finalmente, las olas se cansan de jugar con tu barco y te llevan a unas desconocidas costas de tierras extranjeras. Esta tierra es rica y fértil, como habías soñado que sería; pero como todos los sitios hermosos, está poblada. Importantes caballeros y capitanes reinan sobre los grandes poblados y mandan a sus ejércitos a despejar su país. Podrías someterte al vasallaje de estos hombres y hacer lo mismo con tus seguidores. Pero tú eras rey no hace tanto tiempo y ya has soportado bastantes humillaciones. Serán ellos los que te rendirán homenaje a ti. Obtendrás una corona de nuevo o morirás en el intento. Debes dirigir el poder, como cualquier otro recurso, sabiamente. Aprende a equilibrar tus fuerzas con moderación, y los requisitos del día con las necesidades de los días venideros. Sólo entonces serás el único POWERMONGER.
Claves: 200 territorios a conquistar, cada uno de ellos con una configuración diferente. Estupendos gráficos vectoriales tridimensionales. Opción de dos o más jugadores vía modem. El ordenador puede controlar hasta otros tres ejércitos.



POWERMONGER

• DAYS OF THUNDER
• **Autor: Mindscape**
• **Sistema: Atari**
¡Coge el volante de tu propio coche a 200 kilómetros por hora! A medida que cambias de marcha, verás resbalar a los demás conductores por tu espejo retrovisor... ¿o serás tú el que resbale? Comprueba tus aptitudes como conductor para saber si tienes oportunidad de llegar a Daytona. ¡Da todo lo que puedas y a por el Campeonato! Para poder tomar parte en una carrera debes realizar unas cuantas vueltas de entrenamiento en un circuito vacío para obtener una posición de calificación en la parrilla de salida. A medida que vas de circuito en circuito, tu objetivo final es ser lo suficientemente bueno como para poder correr en la última carrera de la temporada en Daytona, la carrera del campeonato. ¡Buena suerte!
Claves: MIndscape te trae la verdadera emoción de «Days of Thunder» directamente desde la increíble película de Paramount Pictures. Toda la rápida acción de la película llega a tu ordena-



DAYS OF THUNDER

dor. Si tu coche se avería durante la carrera, puedes ir a los boxes a llenar el depósito de gasolina, a reparar los motores o a cambiar los neumáticos.

• TNT

• Autor: Domark

• Sistema: Atari

TNT contiene los cinco juegos más emocionantes que jamás hayas visto: Hard Drivin, Tobbin, A.P.B., Dragon Spirit y XYBots. Hard Drivin es la simulación de coches definitiva en la que se pueden experimentar los riesgos y emociones de unas carreras de coches, en tres dimensiones y con todo realismo. En Tobbin podrás practicar este deporte con los mejores; tendrás que navegar por los rápidos y vivir los desafíos del agua en una aventura llena de acción. APB es un loco juego de policías y ladrones; persecuciones a toda velocidad, peligrosos arrestos y disparos en una alocada historia animada, teniendo siempre cuidado de que no te coja el oficial Bob. En Dragon Spirit tendrás que abrirte camino por los cielos en un frenético vuelo de locura y destrucción; una misión a vida o muerte, derrota o victoria, todo depende de tu habilidad. XYBoots es el más reciente juego de doble acción en pantalla doble; debes mantenerte siempre delante de los mortales robots. Acción sin límites para uno o dos jugadores. Claves: Disfruta del mejor realismo jamás visto y de los estupendos gráficos tridimensionales de TNT.

• THE KILLING GAME SHOW

• Autor: Psignosis

• Sistema: Atari

Ante un público de varios millones de espectadores, el jugador deberá luchar para llegar a lo más alto de las dieciséis fosas de la muerte: fosas infestadas de criaturas artificiales y hostiles, especialmente concebidas por los locos científicos de The Killing Game Show para hacerle las cosas aún más difíciles. Pero el jugador no debe olvidar que el tiempo se va acabando y que el próximo candidato está esperando su turno. Claves: el jugador debe dar al telespectador lo mejor de sí mismo, recogiendo las poderosas armas que se encuentre, ¡si es que puede! La única manera de salir de las fosas es trepar por los muros. Comentario en número sucesivo.

• ARMOUR GEDDON

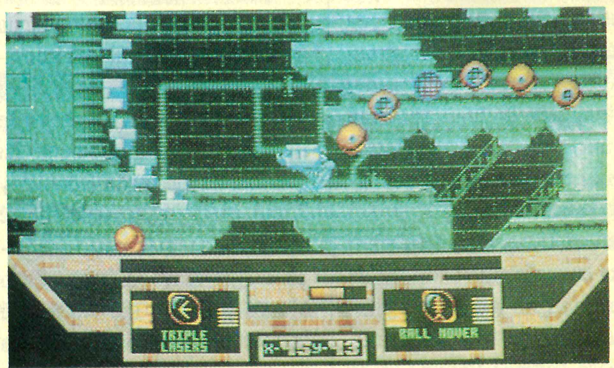
• Autor: Psygnosis

• Sistema: Atari

La naturaleza humana siempre ha sido imprevisible. Un arma de rayos capaz de eliminar toda la vida de la Tierra amenaza ahora a este ignorante planeta. La única esperanza de la raza humana eres tú ... Pero no pienses que la tarea que se te encomienda es imposible: tienes una buena oportunidad de conseguirlo esta vez. A tu disposición se encuentran tanques ligeros, varios aerodeslizadores, tanques pesados, ... además de helicópteros, aviones de caza y bombarderos.

En Armour-Geddon deberás controlar simultáneamente todo este equipo de guerra en un escenario tridimensional utilizando todos los objetos que quedan a tu alcance: rayos láser, granadas, bombas (retardadas y normales), misiles y cohetes. Añade a esto los dispositivos para esconderte, para ver durante la noche, los teletransportadores, los tanques de combustible de reserva y el radar.

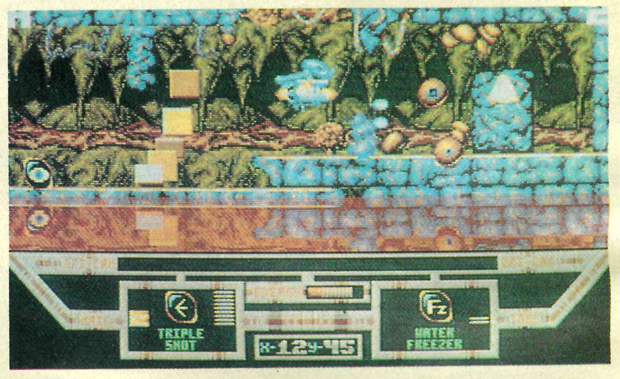
El enemigo no es nada torpe, y su armamento tampoco tiene nada que envidiar al tuyo, por



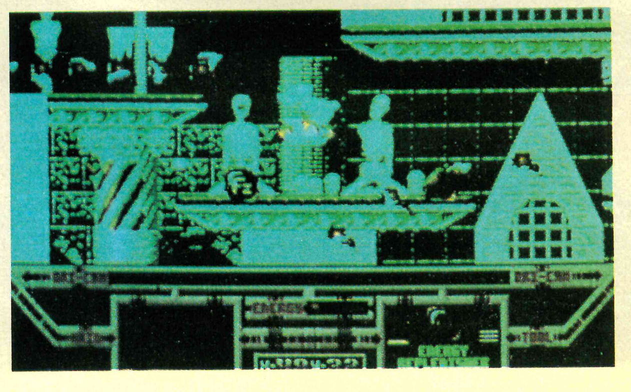
THE KILLING GAME SHOW



THE KILLING GAME SHOW



THE KILLING GAME SHOW



THE KILLING GAME SHOW

suerte, en Armour-Geddon puedes jugar con un amigo si tiene ordenador: ¡dos héroes son siempre mejor que uno!

Claves: con sus gráficos vectoriales de alta velocidad, una zona de ochenta por ochenta kilómetros cuadrados, control real de los vehículos y la posibilidad de enlazar dos ordenadores para jugar simultáneamente la misma aventura, Psygnosis ha conseguido con Armour-Geddon una aventura de simulación única.

• **Autor: Dinamic**
• **Sistema: Atari, PC.**

Durante el mes de Julio, Drosoft distribuirá un nuevo y sorprendente título de Dinamic. Hammer Boy es un juego basado en lo esencial de las máquinas «Hand-Held» de fácil manejo, simples pero altamente adictivas. Este juego se encuentra actualmente licenciado para máquinas recreativas, con la difusión adicional que esto representa.

Hammer Boy es un divertido y frenético arcade de habilidad, en el que tendréis que defenderos a golpe de martillo de la invasión enemiga que os ataca en cada uno de los escenarios.

Una vez que el enemigo o los objetos que nos lancen estén al alcance de nuestro martillo, un sólo golpe bastará para acabar con ellos. El juego consta de cuatro trepidantes fases diferentes ambientadas en épocas distintas de la historia: el Far West, los Mares del Sur, el Castillo Medieval y la Base Espacial. Si conseguimos mantener nuestra fortaleza a salvo durante el tiempo que dure la invasión pasaremos a la siguiente fase, donde deberemos defender una nueva fortaleza. Al pasar de fase podremos conseguir un tiempo extra por cada enemigo que no haya entrado en la fortaleza. Este tiempo se descontará de la duración del siguiente ataque.

Una vez superadas las cuatro fases, estas vuelven a sucederse, pero aún con mayor nivel de dificultad, con lo que Hammer Boy se convierte en un juego imposible de terminar.

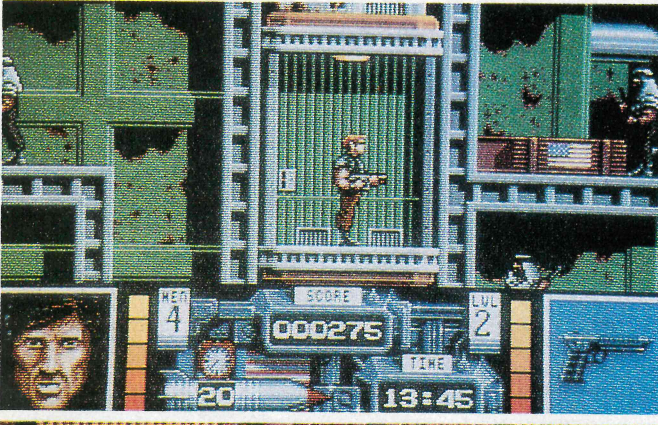
NAVY SEALS

Autor: Ocean
Distribuidor: Erbe

Navy Seals es el primer juego anticipado sobre la guerra del Golfo, puesto que fue concebido bastante antes de tal desastre. Navy Seals (SEa, Air Land) es la historia de un comando de marines cuya misión es destruir unas instalaciones situadas en alguna parte de Irak, reduciendo de paso el mayor número posible de irakíes. Navy Seals (el juego) os propone dirigir un famoso comando compuesto por cinco hombres superentrenados. Estos deambularán (un hombre igual a una vida) a lo largo de ocho niveles que conforman las diferentes misiones. La aventura comienza en los «docks» para continuar en la torre de la comandancia del enemigo. Seguidamente deberéis salvar algunos rehenes antes de caer prisionero. Luego deberás evadirte en dirección a Beirut. Una vez en camino, deberás armarte de coraje y abatir todo aquello que esté en movimiento a tu alrededor para llegar por fin al objetivo de tu misión: la entrada a la nave donde están almacenados los misiles bacteriológicos del «querido» Saddam.

Como podrás constatar es el género de «paseo campestre» que uno hace cotidianamente.

Técnicamente, Navy Seals puede parecer un poco difícil al principio y enfadarte, pero con un poco de perseverancia no dejarás de divertirte. El único punto débil de este juego, es que cuando un marín es herido, se muere y pasas seguidamente a otro (algo rapidísimo). Por el contrario, el scroll, la animación y los gráficos son bastante agradables y hacen de Navy Seals un buen juego arcade.







HYDRA

Autor: Domark
Distribuidor: DROSoft
Sistema: Atari

En el siglo XXI, nada está a salvo de los crueles terroristas que gobiernan los cielos y los mares. Hay que combatir el fuego con fuego. Cuando deben enviarse paquetes especiales de alto secreto, como cultivos de virus mutantes o bombas de alto poder destructivo, que tienen que llegar pase lo que pase a su destino, el único mensajero al que pueden acudir los gobiernos eres tú. Tu nombre secreto: Hydra. En las nueve misiones que se te asignen, el éxito dependerá de tu destreza para dirigir el Hydracraft, utilizando su supervelocidad y su potente armamento para atravesar fuerzas enemigas compuestas por botes, esquíes a propulsión, zepelins, helicópteros, cazas, hovercrafts y otros vehículos. Para hacerte la vida aún más difícil, un mercenario terrorista, «The Shadow», estará escondido en todos los rincones preparado para robar tu precioso cargamento.

Como característica Hydra te proporciona un juego de conducción, vuelo y disparos, rebosante de acción, con una increíble animación sobre asombrosos paisajes digitalizados.

Comentario ampliado en el nº 29 de Atari User, con trucos y ...

No os lo perdáis.



Autor: Psygnosis
Sistema: Atari, PC.

Los lemmings no son estúpidos, simplemente perciben la vida de una forma muy extraña. Por eso la mayor parte de la gente no comprende que se pasen todo el rato paseando de un lado para otro sin tener demasiado cuidado de dónde pisan o caen, aunque pensándolo bien, sí son un poco estúpidos. Pero para eso estás tú: los Lemmings necesitan alguien que les guíe por innumerables pantallas llenas de peligros (peligros para un Lemming, claro). Con más de 100 Lemmings en pantalla, deberás pensar rápidamente qué camino deben seguir para llegar sanos y salvos hasta el final. Cuidado: mientras piensas, los Lemmings continúan caminando incansablemente y, como siempre, sin mirar por donde pisan.

Claves: tú puedes otorgar a los Lemmings pequeñas habilidades que les permitirán trepar, tirarse en paracaídas, construir puentes o cavar túneles. Con más de 100 emocionantes niveles por los que pasar, todos ellos repletos de peligros, estarás salvando Lemmings hasta que ... bueno, hasta que los Lemmings lleguen a casa. La caja de LEMMINGS contiene 100 diminutos adhesivos troquelados en vinilo blanco.

HILL STREET BLUES

Autor: Krysalis

Es el juego más sorprendente de este mes. ¿Recordáis aquella serie policiaca que pasaban por la tele? Canción Triste de Hill Street era su título. Bueno os la recuerdo, a los que la hayan olvidado. Se trataba de una comisaria de policia con sus dramas, crímenes y los perturbadores amores del capitán Furillo. No me sorprendió el saber que este folletín (del que no me perdí ni un capítulo) ha recibido 33 premios diversos y 26 Emmys. Es para reflexionar ¿no?

Comienza la jornada, son las 7 de la mañana y tus hombres responden presente a tu llamada (tú eres el capitán Furillo). Tu objetivo: dirigirles para combatir el crimen en tu viejo barrio de los Angeles. Destacar que el juego está comandado enteramente mediante iconos.

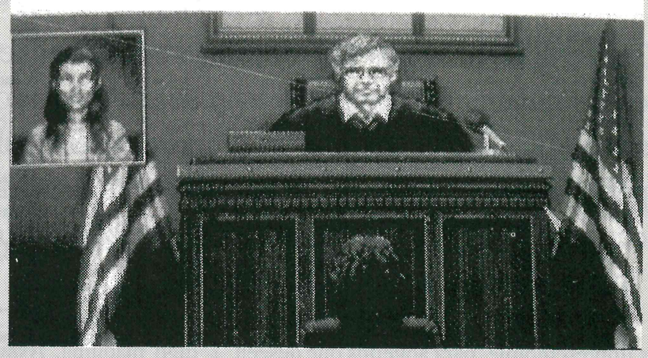
Respecto a la realización, los programadores han tenido la santa paciencia de reconstruir el barrio casa por casa. Además el barrio está vivo, sus habitantes van a sus ocupaciones habituales y cada uno tiene su propia identidad. A nivel de cifras, decir que hay 400 personas y casi el mismo número de coches. Cada personaje posee su propia cara. Algunos están fichados y otros no. A nivel de juego, Hill Street Blues nos recordará a Sim City, un juego interactivo en el cual vosotros debéis intervenir para proteger a vuestros conciudadanos.

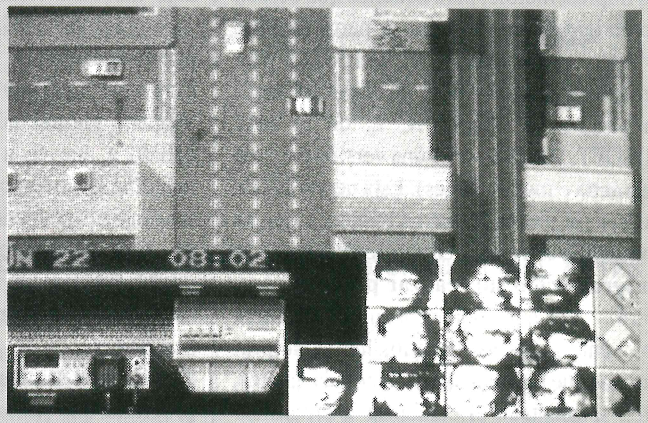
Volvamos a la comisaría: tienes nueve oficiales a tus órdenes. La primera fase consiste en apostarlos en diversos lugares del barrio para evitar cualquier eventualidad. Pueden hacer la ronda a pie o en el coche patrulla, sabiendo que los coches serán más lentos dependiendo de la circulación y de los giros y sentidos prohibidos. Si has optado por el modo "novato", los incidentes no serán muy numerosos (al principio) ni "malignos". El policía entablará un proceso cuando se cometa un acto criminal. Le ponen al tanto del delito, os dan la hora del suceso y la dirección del acusado. Vosotros debéis ir al archivo informático, tomar los datos y antecedentes tanto del presunto acusado como de la víctima. Deberás entonces empezar a moverte por el barrio controlando a todos los implicados en el suceso, algunos huirán al aproximarte, otros conducirán temerariamente. Una vez hayas identificado al culpable hay que prenderle evitando el tiroteo (salvo que sea en legítima defensa). Después más tarde, te citará el tribunal para el proceso. Esto puede parecer simple, pero has de tener en cuenta la velocidad (que lo complica un poco). Los policías tienen otras opciones posibles tales como detener al delincuente, si es un delito menor o bien organizar una barricada si opone una resistencia fuerte. Atención, los habitantes odian los enfrentamientos armados así como las sirenas de la policía, y si saben algo, seguro que no os lo dicen.

Y bueno, para finalizar deciros que Hill Street Blus puede parecer un pequeño juego, pero despues de una hora de jugar, creer que es imposible detenerse.

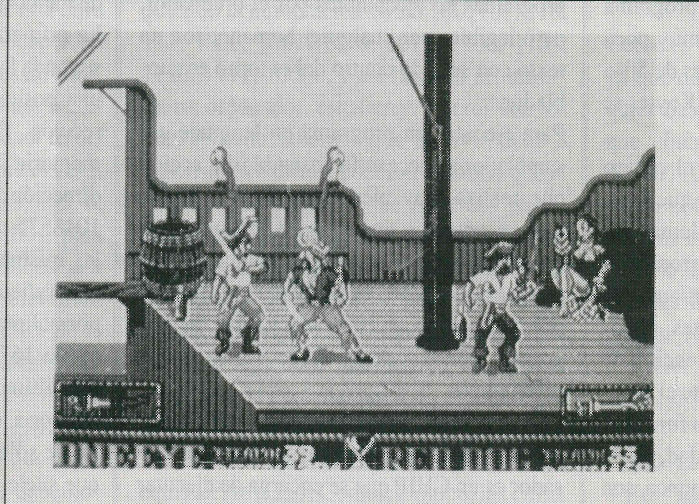
¡Buena suerte!

COUPABLE?, MON CLIENT PLAIDE NON COUPABLE.





SKULL & CROSSBONES



SKULL & CROSSBONES
Autor: Domark
Sistema: Atari, PC

Drosoft y Domark te ofrecen una nueva y original aventura. En ella, encarnas a un pirata que debe abrirse camino a través de los misteriosos y sobrenaturales dominios del Diabólico Hechicero. Deberás pasar por castillos, barcos piratas, una caverna encantada y una isla. Ocho niveles de dificultad, cada uno con obstáculos mortales diferentes.

A lo largo del camino encontrarás cofres repletos de tesoros, copas doradas, sacos de monedas y otros objetos de gran valor que te ayudarán en tu búsqueda. También te encontrarás calaveras y huesos bajo los que se esconden tesoros. Cuantas más cosas de valor recojas, más fácil te será vencer a tus adversarios. Si no recoges tesoros, tu energía se verá reducida y cada vez te será más difícil derrotar a tus enemigos.

Tu misión es recuperar el botín venciendo a los secuaces del Diabólico Hechicero. Mátale y coge el tesoro, pero ten cuidado: puede haber traidores hasta entre tus compañeros.

Prepárate para la batalla final, la batalla contra el Diabólico Hechicero. Derrótale y su poder y riqueza pasarán a tus manos.

Comentario ampliado, trucos, ... en próximo número.

CODIGO MAQUINA - MC 68000 -
(Parte I)

Esta serie de artículos está dedicada a todos aquellos usuarios que deseen sacar el máximo provecho de su ATARI, obligándole a que éste realice las tareas que se le programen. Para comprender y aprender código máquina, a partir de esta serie de artículos, no es necesario saber programar, lo único que hace falta es tener paciencia y mucho tiempo, cuanto más mejor.

Escrito por A. Miguel
Revisión Fernando Perla

Es imprescindible, para aquellos que deseen aprender código máquina a partir de este artículo, que posean un ATARI ST/TT y un programa ensamblador, en especial recomendamos el DEVPAC 2, de la casa HISOFT, y a ser posible una versión original (garantía e instrucciones). Esta necesidad de poseer un ensamblador es debido a que **es imposible aprender cualquier lenguaje de programación sin realizar pruebas.**
Pero... ¿Por qué código máquina? En algún momento todos los usuarios hartos de programas que no se ajustan a la medida, juegos, etc., se plantean el comenzar a programar sus propios programas. Lo normal es comenzar a programar en un lenguaje BASIC, el GFA, HISOFT o cualquier otro. Pero lo normal no siempre es lógico. Un programa escrito en código máquina suele ser unas 100 veces más rápido que él mismo escrito en lenguaje BASIC compilado. Un programa en código máquina suele ocupar muy poca memoria, la reducción de tamaño es de 50 a 1 esto es un programa medio de 50 Kbytes se reduce a un sólo Kbyte.
Siempre existe quien desaconseja el código máquina y normalmente el usuario que sabe, desaconseja su aprendizaje a los demás basándose en que es difícil, que los errores son difíciles de encontrar y que los programas no son transportables. Seamos realistas, el código máquina es el lenguaje más sencillo de todos, el único lenguaje que permite al usuario comprender exactamente cómo funciona un ordenador, y eso de la portabilidad, je, es un bulo, los programas buenos nunca son portables de un ordenador a otro, estén escritos en BASIC, C, Código Máquina, Cobol, Pascal, Forth, ...
Llevar un programa de un ordenador a otro siempre implica reescribir el programa en su totalidad, aunque la estructura del mismo se conserva bien. Una vez visto los pros y los contras pasaremos a intentar explicar lo más claramente posible cómo programar en código máquina. Asumimos que el lector no sabe programar nada y en ningún lenguaje, partimos de cero.
Lo primero que hay que hacer es relacionar lenguaje ensamblador y código máquina, debido a que cada instrucción de código máquina se corresponde a una instrucción en lenguaje ensamblador, y viceversa. A partir de ahora trabajaremos sólo en lenguaje ensamblador, aunque lo denominemos indistintamente código máquina o lenguaje ensamblador.
Programa ensamblador es precisamente el programa que se encarga de transformar un programa escrito en ensamblador a un programa en código máquina. Los programas en código máquina son ejecutables por nuestro ordenador, son una sucesión de números con sentido para nuestro MC68000, mientras los programas en lenguaje ensamblador son programas no ejecutables por el ordenador, pero legibles por cualquier humano, son un texto con sentido dentro del entorno ensamblador.
Para ejecutar un programa en lenguaje ensamblador es necesario ensamblarlo, acción que realiza muy bien el DEVPAC. Ahora que ya tenemos una idea de lo que es un programa escrito en código máquina (lenguaje ensamblador ensamblado), esto es un programa corto, muy rápido y fácil de programar vamos a describir las cosas de que disponemos.
Para nosotros sólo existe el microprocesador y posiciones de "memoria". El microprocesador es un CHIP que se encarga de ejecutar los programas en código máquina, pudiendo realizar operaciones con posiciones de memoria, ya sean operaciones básicas como una suma u operaciones programadas por nosotros como una operación que desplace un Sprite por nuestra pantalla.
Para definir lo que es una posición de memoria, tenemos que pensar enseguida como si la memoria fuese una estantería de muchos pisos (fig. 1). Podemos decir que en cada piso de nuestra estantería caben un máximo de 8 libros. Algunos pisos de nuestra estantería apoyada en la pared, pueden estar agujereados, esto es, cada vez que ponemos libros en ella caen a la habitación de al lado, o bien no ponemos ningún libro y desde la habitación de al lado nos ponen libros.
En nuestro simil estantería/memoria, los libros son bits. Cada ocho bits obtenemos un grupo denominado byte. Las estanterías sin agujero se denominan bytes de memoria RAM y las que tienen agujero bytes de acceso a periférico. Si ponemos libros y caen a la habitación de al lado, son posiciones de memoria de escritura a periféricos, si los libros entran por el agujero se denominan posiciones de memoria de lectura de periféricos.
Hablando ya adecuadamente, las posiciones de memoria del ordenador están numeradas. La primera posición de memoria es la 0, la segunda la 1, la tercera la 2, ... El número de una posición de memoria se denomina dirección. En un 520 existen posiciones de memoria RAM desde la dirección 0 a la dirección 524287, en un 1040 desde la 0 a la 1048575, ... En todos los ATARI ST existen las mismas direcciones de memoria de escritura/lectura de periféricos, estas están normalmente a partir de la posición de memoria 16744448.
Por último existe un tipo de posiciones de memoria, denominada posiciones de memoria de sólo lectura (la ROM del ordenador), que suele oscilar entre los 32 Kbytes en los ordenadores más antiguos y 512 Kbytes en los últimos TT. NOTA: 1 Kbyte= 1024 bytes o posiciones de memoria.
Ahora que conocemos la existencia de un microprocesador (CPU, MC68000 o como queramos llamarle) y la noción aproximada de lo que es memoria del ordenador (posiciones RAM + posiciones de acceso a periféricos + posiciones ROM) es necesario

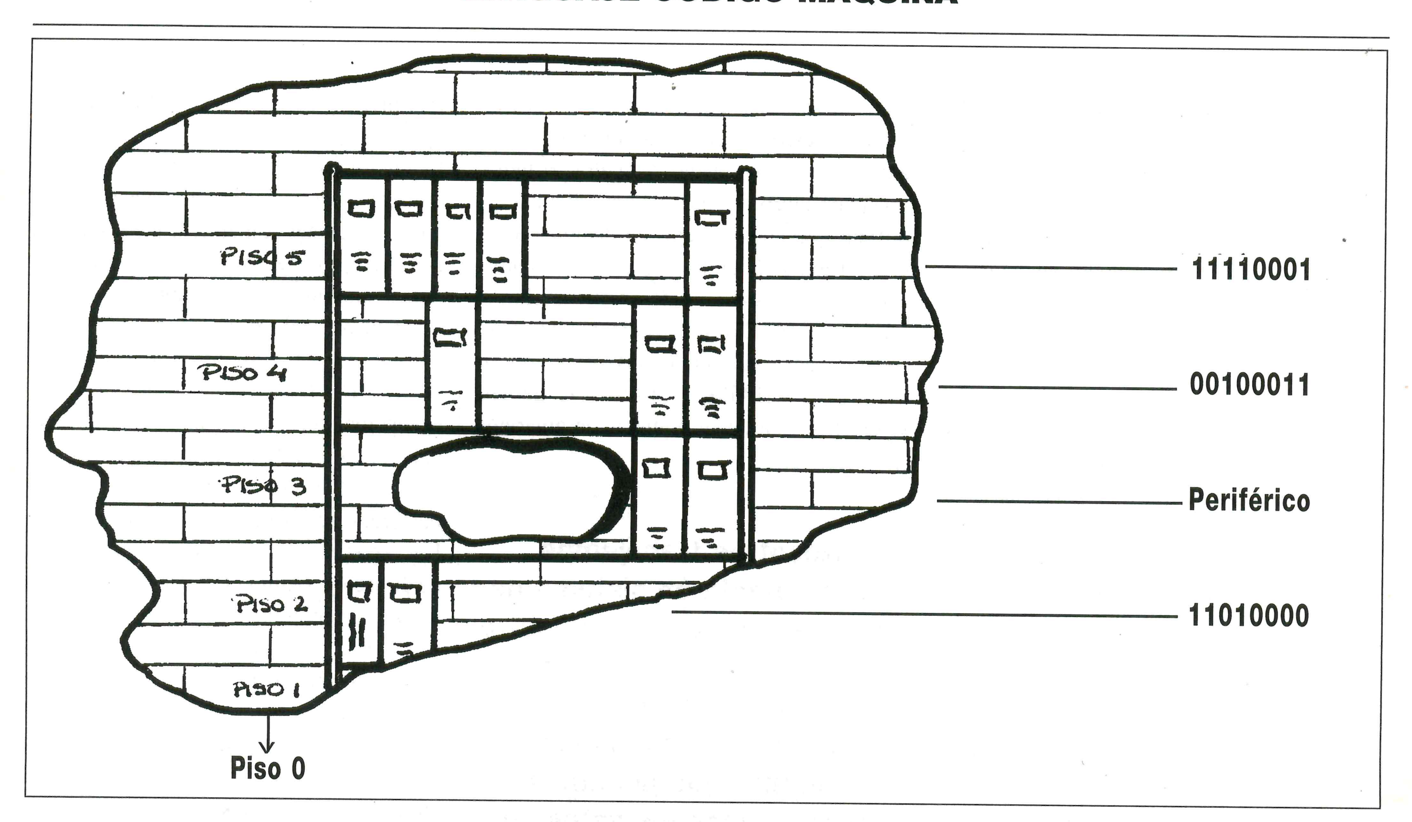


FIG. 1.- ESTANTERIA DE VARIOS PISOS, CON 8 LIBROS POR PISO, ES SIMILAR A UN ORDENADOR DE 6 BYTES DEDICANDO 1 A PERIFERICOS

que estudiemos con detalle lo que es un sistema de numeración binario y un sistema de numeración hexadecimal, ya que de ello depende que aprendamos a programar en código máquina o no.

Nosotros los humanos solemos, normalmente, trabajar en sistema de numeración decimal, esto es que poseemos 10 cifras distintas, siendo la primera el 0, la segunda el 1, la tercera el 2 y así hasta la décima cifra que es el 9. Por un error de concepto en nuestros antepasados de hace miles de años, nosotros normalmente comenzamos a numerar las cosas a partir de 1, pero en informática el primer número es siempre el cero. De este modo numeramos 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, como no tenemos más cifras, para indicar un número mayor utilizamos dos cifras, teniendo la cifra de la izquierda un valor 10 veces superior al de la cifra de la derecha porque estamos en un sistema de numeración de 10 cifras. De este modo nuestra numeración continúa 10, 11, 12, 13, 14, 15, 16, ..., 98, 99. Aquí en 99 nos topamos con el mismo problema, que debemos solucionar con el mismo truco de antes, esto es utilizando tres cifras, teniendo siempre la de la izquierda un valor 10 veces mayor que la del centro y esta a su vez un valor 10 veces superior a la de la derecha. La numeración sigue 100, 101, 102, 103, ...

Ahora tenemos un sistema de numeración binario, o sea, de tan sólo dos cifras, el 0 y el 1. Si comenzamos a numerar, tendremos o, 1, y el problema, necesitamos una cifra más. Por ello, al igual que en el sistema de numeración decimal añadimos una cifra a la izquierda, aunque esta vez su valor respecto a la de la derecha es tan sólo 2 veces superior por estar en un sistema de numeración binario. Así nuestra numeración continúa 10, 11, y otra vez problema, problema que solucionamos con el truco de la tercera cifra, siguiendo la numeración como 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111, 10000, y así hasta el infinito.

En un ordenador, estos unos y ceros son los bits, y como sabemos que un byte tiene 8 bits, esto es 8 cifras, podemos deducir que el número más pequeño que podemos representar es 00000000 y el más gordo 11111111. Si juntamos varios bytes, podemos automáticamente tener más cifras. Para convertir un número de binario a decimal sólo debemos hacer lo siguiente: multiplicar la primera cifra (la de la derecha) por 1, la siguiente por 2, la siguiente por 4, la siguiente por 8, ...

De este modo descubrimos que 11111111 es equivalente a 255. Como es lógico, las cifras de cualquier número, ya sea en decimal o binario, están siempre numeradas, siendo la de la derecha el número 0, la siguiente la 1, la siguiente la 2, ... Esta numeración cobra un poco más de sentido si tenemos en cuenta que cada cifra tiene un valor VxS^L, siendo V el valor de la cifra, 0, 1, 2, 3, ..., S= 2 si el sistema es binario, S=10 si es decimal, S=16 si es hexadecimal y L el lugar que ocupa la cifra. De este modo podemos asegurar que 00100110 representadoen binario es:

```
 7   6   5   4   3   2   1   0 <- valor de L
+---+---+---+---+---+---+---+---+
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
+---+---+---+---+---+---+---+---+
```

$0x2^0 + 1x2^1 + 1x2^2 + 0x2^3 + 0x2^4 + 1x2^5 + 0x2^6 + 0x2^7 = 38$

Como es lógico 38 es igual a $3x10^1 + 8x10^0$, ya que debe cumplirse esta regla para cualquier base en que trabajemos. El resultado que obtengamos puede ser 38 a partir de 00100110 o bien 00100110 a partir de 38 según operemos en binario o en decimal al realizar la suma de los VxS^L. Operar en un sistema u otro se rige siempre bajo las mismas normas. Por ejemplo si deseamos multiplicar 1001 x 110 en binario deberemos:

```
  1001
 x 110
 ------
  0000  La cifra 0 de 110 por cada una de las de 1001.
 1001   La cifra 1 de 110 por cada una de las de 1001.
+1001   La cifra 2 de 110 por cada una de las de 1001.
 ------
110110  Sumamos columna con columna, tal y como
        lo haríamos en base decimal.
```

Como véis el resultado de 6x9 (1001x110) es igual a 54 (110110). Sencillo, ¿no? Pues lo mismo se aplica a cualquier operación.

Todos os habréis dado cuenta de que el

sistema binario no es un sistema adecuado de numeración para los humanos a pesar de que sea el que utiliza internamente un ordenador, dado que cualquier número que se aleje demasiado de 0 tomará tamaños relativamente grandes e incomodos de manejar. Por ejemplo, para representar el año en que estamos, 1991, en binario tendremos que escribir 11111000111.

Por otra parte un sistema de numeración decimal no es adecuado para las computadoras, ya que las operaciones son muy complejas, multiplicar requiere una tabla con 121 valores en un sistema de numeración decimal, mientras que en un sistema binario con 4 valores bastan. Lo mismo es válido para la suma, resta, ... Por añadidura, la electrónica sólo acepta dos estados, 1 y 0 (pasa corriente o no pasa), obligando a todas las cifras a poseer valores inferiores a 2. Por definición un byte tiene 8 cifras, pudiendo tomar un valor entre 8 y 255.

Si deseamos juntar dos bytes para obtener 16 cifras y así poder representar un número mayor, en binario el byte 1 podría tener un valor 00000111, mientras el byte 0 podría tener un valor 11000111. Juntando los dos bytes obtenemos 0000011111000111 esto es 1991. Si hacemos lo mismo en decimal el byte 1 tiene un valor 007 y el byte 0 un valor 199, juntando los dos obtenemos 0007199, que no es bajo ningún caso 1991. Unir dos bytes en decimal, implicaría hacer 7x256+199=1991.

Como internamente hagamos lo que hagamos el ordenador va a trabajar en binario, debemos buscar algún sistema de numeración que más o menos sea compatible con él y que nos permita aclararnos. Este sistema de numeración es el llamado sistema hexadecimal, esto es un sistema de numeración de 16 cifras, que numera de la siguiente forma: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, ... hasta el infinito.

Con este sistema de numeración un byte siempre tiene 2 cifras en vez de 8. Podemos simular el «bitón». Un «bitón» son 4 bits, que se puede representar con una sóla cifra en hexadecimal, sea cual sea su valor y en informática se denomina nibble. (1 byte = 2 nibbles = 8 bits). Si ponemos un 07 en el byte 1, y un C7 en el byte 0, podemos juntarlos y crear 07C7, cuyo valor en decimal es $0x16^3 + 7x16^2 + Cx16^1 + 7x16^0 = 1991$.

Debemos tener en cuenta que A, B, C, D, E y F son cifras añadidas que utilizamos para representar 10, 11, 12, 13, 14 y 15 en una sóla cifra. Aconsejamos al lector interesado que experimente lo máximo posible realizando a mano operaciones numéricas con los distintos sistemas, así como pasar números de uno a otro. Es recomendable, para ahorrar tiempo en un futuro disponer de una calculadora normal que pueda operar en cualquiera de estos sistemas. Esta se puede encontrar como programa accesorio, como en el Harlekin o bien como aparato independiente. (Sin ánimo de hacer publicidad las calculadoras científicas SHARP, a diferencia de las CASIO, suelen llevar numeración binaria y hexadecimal).

Nosotros trabajamos normalmente en sistema de numeración decimal, nuestro ordenador siempre trabaja en sistema de numeración binaria. Por comodidad siempre que podamos escribiremos los datos en decimal, si no es posible lo haremos en hexadecimal, y por último lo haremos en binario si no queda otro remedio.

Insistimos, todos los sistemas de numeración son equivalentes, simplemente cambian su forma en el papel, pero todas sus operaciones son válidas. Nosotros trabajamos normalmente en sistema de numeración decimal, nuestro ordenador siempre trabaja en sistema de numeración binaria. Por comodidad siempre que podamos escribiremos los datos en decimal, si no es posible lo haremos en hexadecimal, y por último lo haremos en binario si no queda otro remedio.

Ya conocemos algo más de nuestro ordenador: tiene microprocesador, posiciones de memoria y trabaja en sistema de numeración binario, aunque por comodidad podemos asumir que trabaja en hexadecimal, esto es a 2 bitones (nibbles) por byte. Ahora continuaremos estudiando un poco los recursos que tiene el microprocesador.

Como ya dijimos antes las posiciones de memoria RAM contienen un byte que puede tener un valor entre 0 y 255, valor que podemos leer y escribir a voluntad como veremos más adelante. Si juntamos 2 bytes podemos representar un número entre 0 y 65535 y si juntamos 4 bytes podemos representar números entre 0 y 4295032831. A dos bytes unidos se denomina palabra, y a la unión de dos palabres se denomina palabra larga. De este modo 1 palabra larga = 2 palabras = 4 bytes = 8 nibbles = 32 bits.

El microprocesador internamente tiene 19 registros. Los registros se pueden imaginar como posiciones de memoria que están dentro del microprocesador, y que no están numerados, sino que tienen nombres propios.

Los registros del 68000 son PC, A0, A1, A2, A3, A4, A5, A6, USP, SSP, D0, D1, D2, D3, D4, D5, D6, D7 y SR. Todos los registros a excepción del SR pueden almacenar una palabra larga, esto es, tienen 32 cifras o sea 4 bytes. El SR tan sólo puede almacenar una palabra, o sea, 16 cifras, es decir, 2 bytes.

Los registros A0-A6 se denominan registros de dirección, (Adress Register), los registros D0-D7 se denominan registros de datos (Data Register). El registro PC se denomina contador de programa (Program Counter), el registro SR registro de estado (Status Register) y por último USP y SSP son el apuntador a la pila de usuarios y el apuntador a la pila supervisor (User Stack Pointer & Supervisor Stack Pointer).

Una vez definidos lo que son registros, vamos a ver que funciones desempeña cada uno. Los registros de dirección, esto es, la dirección a una posición de memoria, mientras que los registros de datos, almacenan datos, cualquiera que se nos ocurra. Estos dos tipos de registros a la hora de la verdad pueden ser utilizados por el programador de la forma que estime conveniente.

Los otros cuatro registros tienen asignada «de fábrica» su función. El contador de programa, es un registro de dirección muy especial que contiene la dirección a la posición de memoria que contiene la próxima instrucción a ser ejecutada por el microprocesador. Por ejemplo si el valor del PC=45892, el microprocesador leerá el valor que hay almacenado en dicha posición de memoria, lo interpretará adecuadamente e incrementará el contador de programa. El contador de programa debe contener siempre un número binario de 32 cifras, siendo su cifra de la derecha cero, debe ser siempre un número par, debido a que las instrucciones en código máquina ocupan siempre una palabra, dos bytes.

Veremos en un futuro, una vez que comencemos a programar, más características del contador de programa.

El registro de estado, nos indica el estado en que se encuentra el microprocesador, según determinadas cifras del SR estén a 1 o a 0 nos indicará estado. Por ejemplo, si el microprocesador realiza una comparación de dos elementos iguales, automáticamente la cifra 2 del registro de estado adquiere valor 1,

mientras que si son distintos valor 0. Veremso más a fondo las características de este registro en el próximo capítulo, a medida que vayamos programando.

Por último nos quedan los apuntadores a la pila de usuario y a la pila de supervisor. La pila nos la podemos imaginar como un pinchapapeles colgado del techo (ver fig. 2). En él podemos meter datos, esto es pinchar papeles, de forma que cuando deseemos extraerlos habrá que hacerlo de modo inverso al que lo hemos metido. O sea, el último papel en entrar es el primero en salir.

Como la memoria del ordenador no posee ningún pincho que obligue a los datos que entren primero salgan los últimos y viceversa, sino que más bien sigue la estructura de las estanterías, para recrear la estructura de pila necesitamos la ayuda de un registro de dirección. Este apunta a una dirección de memoria, por ejemplo supongamos que apunta a la dirección 100 y deseamos meter en la pila los siguientes números: 45, 89 y 1228, que son palabras (nunca se pueden meter bytes sueltos en la pila). Para ello restamos al puntero de la pila dos unidades, toma valor 98 y metemos en la dirección de memoria 98 el valor 0 y en la posición de memoria 99 el valor 45, con el siguiente dato hacemos lo mismo, restamos dos unidades, esto es que toma valor 96 y metemos en la posición de memoria 96 el valor 0, y en la posición de memoria 97 el valor 89 y por último para introducir otro dato restamos otras dos unidades, toma valor 94, e introducimos en la posición de memoria 94 el valor 4 y en la posición 95 el valor 204 (si no se entiende repasar la explicación sobre sistemas de numeración).

Para extraer datos de la pila realizamos el proceso inverso, primero leemos la posición de memoria apuntada por el puntero de pila, las posiciones 94 y 95, con lo que extraemos 1228 y sumamos al puntero de pila dos unidades, dejándolo con valor 96.

Si deseásemos introducir una palabra larga, esto es 4 bytes tendríamos que restar 4 unidades en vez de 2. El proceso parece complicado, pero a la hora de programar es tan sencillo como:

```
move.w  #45,-(A7)
move.w  #89,-(A7)
move.w  #1228,-(A7)
```

instrucciones que como veremos en el próximo capítulo indican al procesador mover una palabra (move.Word), en concreto el dato #45, #89 y #1228 al lugar apuntado por A7, esto es (A7), restando dos unidades antes de realizar el movimiento, esto es -(A7). Para extraer el método es el mismo:

```
move.w  (A7) +,D0.
```

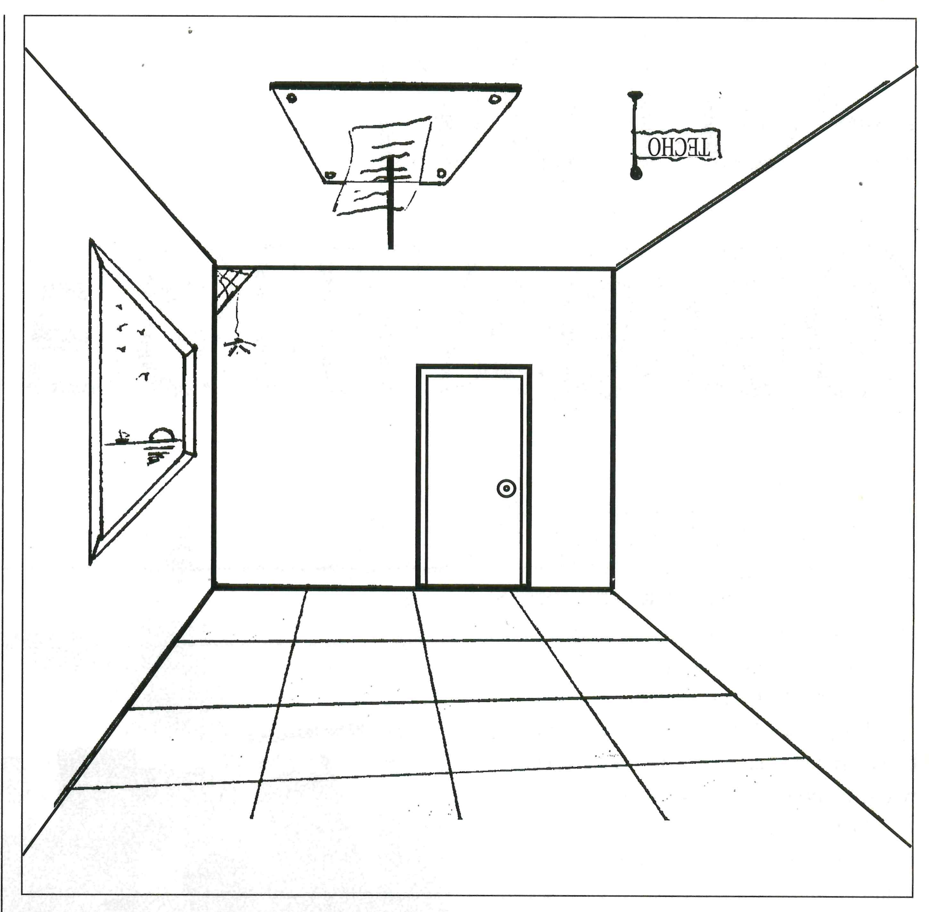


FIG. 2.- PINCHA PAPELES COLOCADO DEL TECHO, SU FUNCIONAMIENTO ES IDENTICO AL DE UNA PILA DE DATOS

Esta instrucción mueve una palabra desde el lugar apuntado por A7 hacia el registro D0, incrementando posteriormente el valor de A7 en dos unidades. Si no lo entendéis, no es grave, ya que explicaremos con más detalle el funcionamiento exacto más adelante.

Muchos ya os habréis dado cuenta de la ventaja que ofrece guardar datos en la pila en vez de guardarlos en posiciones de memoria fijas. Con el sistema de pila, no hay que recordar el lugar en que se encuentra cada dato, basta saber que está en la pila, que es el último dato introducido, el penúltimo, el antepenúltimo, el ... Este sistema de guardar datos es muy utilizado, tanto por nosotros como por el ordenador. Como todos sabéis, si nosotros realizamos un programa, mientras se esté ejecutando, simultáneamente el ordenador ejecuta el suyo, para poder mover el ratón, mirar si llega algo por el MODEM, ... Este programa del sistema mete datos en la pila, datos que podrían interferir con los nuestros. Por ello el sistema operativo (el TOS, GEM o como deseemos llamarlo) posee un puntero de pila propio llamado pila supervisor mientras nosotros los usuarios disponemos del puntero de pila usuario. De este modo queda invalidada la posibilidad de que ambas pilas se interfieran, algo así como si el apuntador a la pila usuario tiene valor 100 y el apuntador a la pila supervisor tiene valor 200.

Como curiosidad, el bit 13 (cifra 13) del SR toma valor 1 cuando estamos en modo supervisor, esto es hacemos de sistema, y valor 0 cuando estamos en modo usuario. Nosotros como usuario podremos en circunstancias especiales pasar a ser sistema (supervisor), pudiendo así realizar cosas que no podemos hacer en modo usuario, como escribir en SR, que veremos más adelante.

Confiamos en que esta primera parte del artículo haya sido del agrado de todos y que en el próximo número estéis todos preparados con vuestros programas ensambladores y dominéis el sistema de numeración binaria y hexadecimal.

PSION ORGANISER II

Este mes vamos a comentar un periférico que puede conectarse a nuestro ATARI y que es algo más que un simple periférico, se trata del Ordenador de Bolsillo PSION ORGANISER II. Nació hace ya unos cinco años, pero gracias a sus nuevas versiones no ha quedado desfasado.

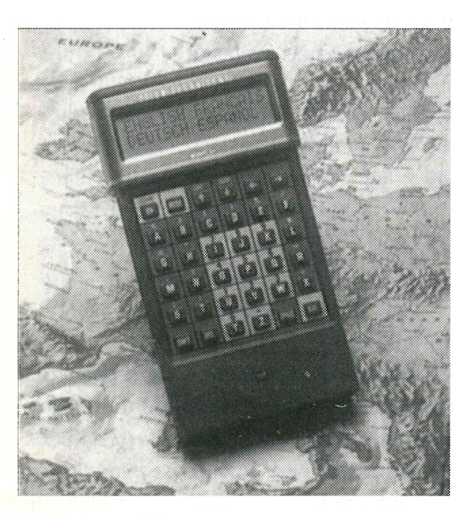
En primer lugar vamos a analizar sus características para posteriormente ver como puede complementarse con nuestro ATARI.

Primero hay que tener en cuenta que el Psion es un ordenador de bolsillo, cuyo tamaño es un poco mayor que el de una cajetilla de tabaco estándar y que tiene en principio todas las características de un ordenador de sobremesa.

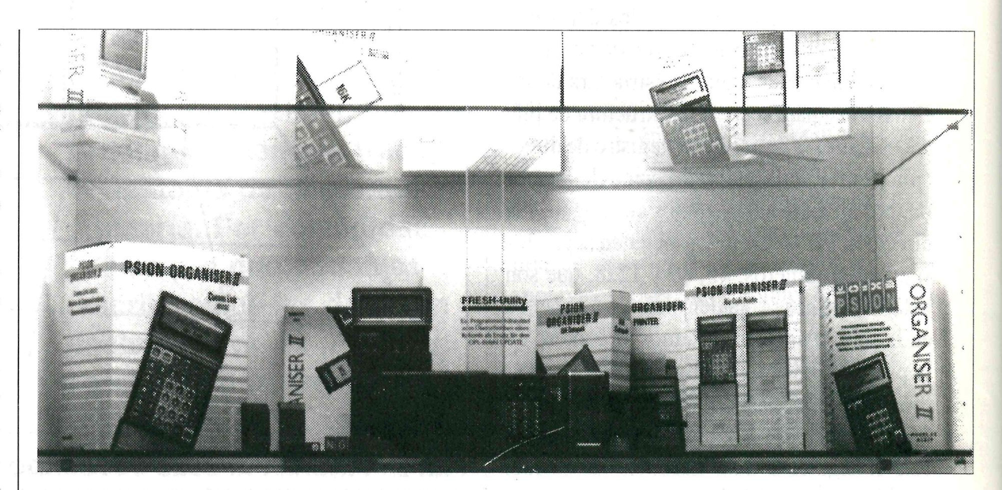
Incorpora una CPU compatible a nivel de código con la CPU de 8 bits MC68000 y funciona bajo una frecuencia de reloj de unos 2 MHz.

Existen dos modelos de Psion compatibles entre sí, el modelo CM y el modelo XP.

El primero, CM, dispone de 8 Kbytes de memoria interna y 32 Kbytes de ROM con funciones incorporadas de agenda, archivador, calculadora, reloj, alarma y OPL. Puede aceptar como medio de almacenamiento externo dos DATAPACKs de 64 Kbytes cada uno, pudiéndose por tanto disponer de hasta 128 Kbytes de almacenamiento externo. Este modelo dispone de una pantalla de cristal líquido de 2 líneas.



PSION modelo CM



Escaparate de productos PSION donde se puede apreciar el modelo XP de 64 K

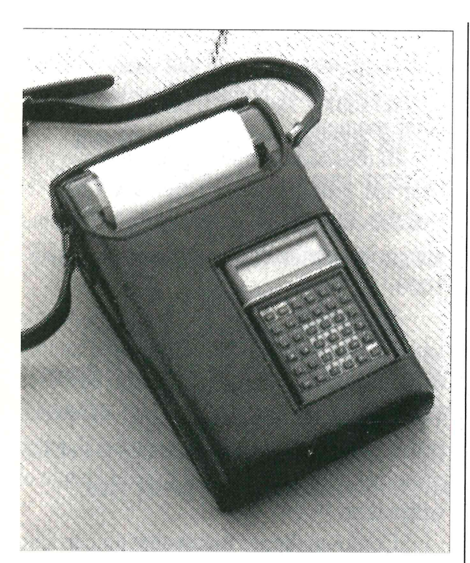
El segundo, el XP, añade a las características del CM una memoria interna de 4 a 8 veces mayor, la posibilidad de añadirle DATAPACKs de 128 Kbytes y RAMPACKs de 32 Kbytes, hoja de cálculo de bolsillo, lector de código de barras y lector de tarjetas magnéticas. El modelo XP puede funcionar en cuatro idiomas distintos, y en la versión castellana dispone de Castellano, Italiano, Inglés y Alemán. La pantalla de este modelo es de 4 líneas. Cada versión tiene cuatro idiomas distintos, y si sumamos todos los idiomas de todas las versiones nos encontramos que podemos tener hasta 11 idiomas diferentes.

En los párrafos anteriores se han mencionado los DATAPACKs y los RAMPACKs. Estos son dispositivos de almacenamiento externo, que suplen la falta de diskettes. El DATAPACK es un dispositivo de lectura escritura, en el cual no se pueden borrar los datos, y por tanto no se pueden modificar. La idea de los DATAPACK es que vamos escribiendo datos, con posibilidad de tacharlos, hasta que el dispositivo está lleno, momento en el que debemos borrarlo. Para borrar un DATAPACK necesitamos de un borrador de EPROMs (un tubo fluorescente ultravioleta), ya que este es simplemente eso, una EPROM.

El otro dispositivo mencionado es el RAMPACK, un dispositivo de lectura y escritura con posibilidad de borrado, y consiste en una memoria RAM estática con una pequeña batería de Litio. Los RAMPACK actualmente comercializados en España disponen de tan sólo 32 Kbytes de memoria, pero existen en el extranjero RAMPACKs de 256 Kbytes, pudiendo entonces disponer en el bolsillo de un ordenador de 578 KRAM.

El Psion está pensado para uso profesional, a pesar de que disponga de algunos juegos de dominio público. Por ello para el Psion existen un buen surtido de periféricos como por ejemplo la Psion Printer II.

Los DATAPACKs y los RAMPACKs son dispositivos de almacenamiento externo, que suplen la falta de diskettes.



PSION Organiser con PRINTER II en una funda protectora

Este periférico consiste en una impresora portátil, de carro pequeño con 80 columnas en modo comprimido que utiliza papel térmico de bajo coste. Otros periféricos comunes son el lector de códigos de barras y tarjetas magnéticas que permiten eso, leer barras y tarjetas abriendo al Psion un abanico de posibilidades en el mundo profesional que abarca desde supermercados hasta control de seguridad.

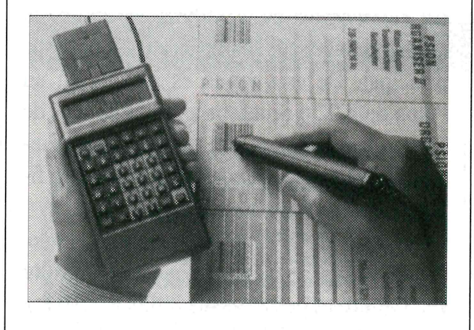
También dispone de otro grupo de periféricos muy específicos como pueden ser un radio transmisor, que permite al Psion recibir y enviar datos a una velocidad de 1.200 baudios por radio (alcance de 3 a 30 Km según modelo) a un ordenador de sobremesa tipo ATARI ST o TT. En concreto este es el sistema utilizado por la policía municipal de Barcelona para ver si compensa o no compensa retirar un vehículo mal aparcado. Para ello, el policía a través del Psion y la matrícula del vehículo infractor solicita a la central si dicho vehículo está en busca y captura (esto es si debe ya dinero) en tal caso el policía solicita a la central que acuda una grúa a retirar el vehículo, y en el caso contrario se limita a colocarle en el limpiaparabrisas un boleto de denuncia.



PSION Organiser II con un paquete para controles de horarios de trabajo en empresas con mucho personal

El Psion dispone de programas ya hechos, como el Topack que ayudan al profesional dedicado a la topografía en las tareas de campo más monótonas y reiterativas, leyendo los datos directamente de las libretas taquimétricas de los taquímetros Tagman, Celta 4 y FC2DIS. Una vez realizadas las mediciones del terreno, existe la posibilidad de sacar un mapa topográfico a partir de los datos tomados mediante la ayuda de un plotter.

Para el Psion existe todo tipo de programas, programas que van desde el control de horarios de los empleados de una fábrica, hasta facturación y control de Stocks, pasando por programas de astronomía, química, matemáticas, cálculos energéticos, dietética,...



PSION con lector de códigos de barras

El Psion, al tener un aspecto similar al de una calculadora de bolsillo convencional, puede utilizarse impunemente en cualquier examen, ya sea de Bachiller, Formación Profesional o Universidad, pudiendo así almacenar todo tipo de fórmulas y anotaciones que no son posibles recordar a última hora (chuletas). Incluso es posible, sin dificultad, proteger el Psion por clave, a fin de que los profesores curiosos no puedan encontrar dentro del Psion las chuletas (ocultando ficheros o programas que sólo salen si previamente se escribe una clave).

Una vez que hemos analizado por encima las posibilidades del Psion, vamos a analizar ligeramente las aplicaciones que existen en su ROM. En estos párrafos vamos a referirnos siempre al modelo XP de 32 Kbytes internos que debido a su precio de 30.000 pesetas dispone de una calidad/precio óptima, ya que el modelo CM cuesta la mitad pero es mucho peor, mientras que el modelo 64 Kbytes cuesta 10.000 pesetas más y en raras ocasiones son necesarios más de 32 Kbytes.

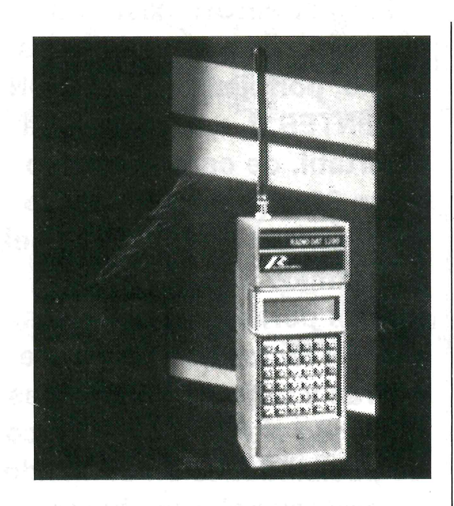
Para el PSION existen un buen surtido de periféricos como, por ejemplo, la PSION PRINTER II, una impresora portátil, de carro pequeño con 80 columnas en modo comprimido que utiliza papel térmico de bajo coste. Otros periféricos comunes son el lector de códigos de barras y tarjetas magnéticas abriendo al Psion un abanico de posibilidades en el mundo profesional que abarca desde supermercados hasta control de seguridad.

Cuando arrancamos el Psion, nos aparece en pantalla una serie de opciones, que podemos seleccionar pulsando su letra inicial o con los cursores (como si de un GEM arcaico se tratara). Estas opciones, son las que llaman a los distintos programas que dispone el ordenador en la ROM, programas de agendas, alarmas, notas,...

El funcionamiento del ordenador se puede enfocar desde varios puntos de vista, pero personalmente creo que la forma más adecuada es dividirla en 6 partes, esto es programas de gestión de la máquina, idioma-estadísticas de los discos (DATAPACKs y RAMPACKs) -reloj-cronómetros-alarma-temporizador, agenda, gestión de ficheros, calculadora y lenguaje de programación OPL.

El primer grupo de utilidades nos permite ver qué espacio del disco está libre, qué porcentaje del sistema ocupa la agenda, el libro de notas, los programas, etc. También incluyo en este grupo la posibilidad de programar hasta 10 alarmas diferentes, que son programables con fecha y hora, hacerlas diarias, semanales, ..., pudiendo por ejemplo programar el Psion para que de lunes a viernes suene a las 8 de la mañana, los sábados a las 10 y los domingos que no suene.

Se dispone también de cronómetros con Lapso y 1 centésima de precisión así como temporizador con cuenta atrás. Una de las



PSION conectado a un emisor/receptor de radio

opciones interesantes es la opción de horario internacional, con lo cual es posible disponer de cualquier horario del mundo. Para ello, inicialmente le decimos al Psion que estamos en una ciudad con horario español (Madrid, Barcelona, Ceuta, Valladolid, ...) y lo ponemos en hora. En este momento, en el que el Psion sabe en que lugar nos encontramos, le podemos solicitar el prefijo telefónico para llamar por teléfono a cualquier ciudad del mundo, ya sea nacional o internacional. Si cambiamos el lugar en que se encuentra el Psion, por ejemplo LIMA, todos los prefijos se ajustarán de acuerdo con la compañía telefónica local, así como la hora.

En la opción agenda, dispone de una retícula de datos en donde están todas las horas y días de nuestra era, en ella podemos ir apuntando cosas, por ejemplo exámenes, citas, etc., dándole al Psion la posibilidad de avisarnos cuando el acontecimiento suceda. En fin se trata de una agenda electrónica que simula a los diarios de las agendas convencionales en papel.

La opción de gestión de ficheros, nos permite lo que permite cualquier base de datos, añadir fichas, ordenar por un campo, sumar items, ... Hay una pequeña diferencia que hace al Psion ligeramente diferente, que es la posibilidad de que cada ficha tenga un número de campos distintos, pudiendo así tener en un fichero fichas que tienen cierta relación, pero que no tienen nada que ver entre sí.

Una opción interesante es la calculadora que incorpora el Psion, con posibilidad de trabajar con funciones científicas y funciones definidas, así como con 10 memorias.

Por último, la opción que hace superior el Psion a otros ordenadores de la competencia, es el OPL un lenguaje de programación estructurado y sin número de línea que debe ser compilado antes de ser ejecutado. O sea, el Psion Organiser II es el único ordenador de bolsillo conocido que incorpora compilador.

El OPL es un lenguaje de alto nivel que está a caballo entre el BASIC convencional y el lenguaje C. Para programar en OPL, vamos definiendo procedimientos y funciones que se van llamando unos a otros, pudiendo existir variables locales y globales, así como pasar/devolver parámetros a/de las funciones.

Cada función deberá ser escrita en el editor de OPL y posteriormente compilada por separado. Normalmente las funciones compiladas suelen tener siempre menos de 500 bytes, pudiendo así tener una amplia biblioteca de funciones. Cuando ejecutamos un programa, o sea una función que llama a otras funciones, se carga dicha función a la memoria principal del ordenador, tanto si está en la memoria interna como si está en memoria de almacenamiento externo. Si esta función llama a otra se carga esta otra en memoria, y así hasta que una función termine, momento en que se libera espacio de la memoria, borrando la función de la memoria principal. De este modo es posible en un Psion de 32K ejecutar programas de más de 100K, para ello hay que evitar que las funciones sean muy grandes. (Ejecución de ficheros en Batch).

Al ser el OPL un lenguaje compilado, no solamente se consigue que el ordenador sea rápido, (a pesar de sus 8 bits a 2 MHz), sino que se optimiza la memoria, dado que cada programa se compone de dos ficheros. Un fichero contiene el listado del programa, y otro contiene el código ejecutable del programa, pudiendo borrar el listado en cualquier momento.

La opción que hace superior el Psion frente a otros ordenadores de la competencia, es el OPL, un lenguaje de programación estructurado y sin número de línea que debe ser compilado antes de ser ejecutado.
El Psion Organiser II es el único ordenador de bolsillo conocido que incorpora compilador.

El resultado es que se pueden comercializar los programas sin tener que ceder el listado, cosa que no es posible con ordenadores de bolsillo tales como la casio PB-1000.

Para llamar a las funciones, podemos hacerlo desde el módulo de programación, opción ejecutar función, o bien desde la calculadora, como si de otra función se tratase, o por último añadir la función al menú principal que aparece cuando se enciende el ordenador.

Como cualquier ordenador no cerrado, el Psion puede programarse en código máquina para realizar funciones que no dispone el ordenador y que en OPL serían quizá difíciles de programar, como por ejemplo la ampliación a OPL de Demus & Rathe que permite crear ficheros indexados. Aunque la programación en código máquina está desaconsejada, ya que de colgarse el equipo habría que arrancarlo de nuevo perdiendo todos sus datos.

Para aquellos que deseen sacarle el máximo partido, o que deseen que forme parte de un proyecto, pueden solicitar junto a su Psion un manual de sistema operativo de 800 Kbytes en disco, en el cual se explica detalladamente su sistema operativo.

Al Psion se le puede añadir, opcionalmente, un puerto de serie que lo conecta a nuestro ATARI, puede así transmitir datos a éste, ya sea para guardar ficheros o para leerlos.



PSION conectado a un taquímetro

Al Psion se le puede añadir, opcionalmente, un puerto de serie que lo conecta al ATARI, pudiéndose así transmitir datos a éste, ya sea para guardar ficheros o para leerlos.
De este modo, se nos permite pasar al Psion datos desde el Superbase, o textos desde nuestro editor favorito sin ningún problema.

De este modo podemos pasar al Psion datos desde el Superbase, o textos desde nuestro editor favorito sin ningún problema.

También disponemos de la posibilidad de imprimir cosas desde el Psion a la impresora de nuestro ATARI con un pequeño programa SERVER. De este modo queda enlazado nuestro Psion con el ordenador. El interface RS232C del Psion incorpora una pequeña EPROM con una serie de funciones que nos permiten utilizar el Psion como terminal, enviar y recibir datos desde el ordenador bajo los protocolos Psion, ASCII o XMODEM. La configuración que hemos probado nosotros es el Psion comunicado por XMODEM a través del programa de comunicaciones FLASH.

Pero no es oro todo lo que reluce, el Psion dispone de un par de defectos a considerar, en primer lugar, su teclado. Cuando uno estrena el Psion lo primero que molesta es su teclado, que no es de tipo QWERTY sino de tipo ABCDE, lo cual en principio no parece grave.

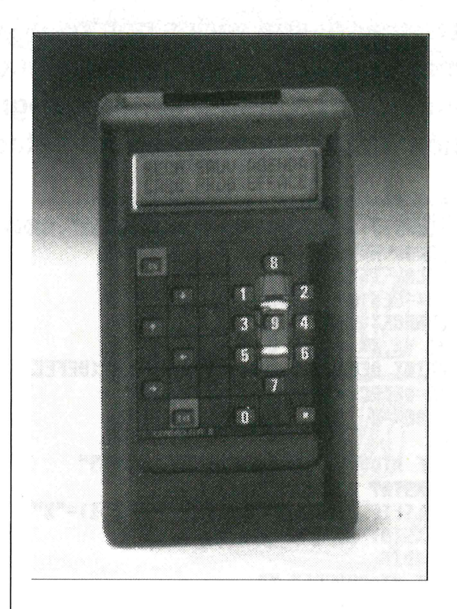
Profundizando un poco, observamos que debajo del abecedario está el teclado numérico, es decir, las mismas teclas que sirven para que introduzcamos letras nos sirven para introducir números. Para ello el Psion dispone de dos modos de cursor, una rallita parpadeando o un cursor completo, indicando que el teclado está en modo abecedario o en modo numérico. Por suerte todos los módulos del Psion están bien programados y siempre aparece en pantalla el modo adecuado, siendo escasas las veces que hay que cambiar manualmente. Por otra parte un usuario normal, acaba acostumbrándose a este teclado, pudiendo al final teclear a velocidades razonables (similar a la que se teclearía en cualquier otro teclado).

Un defecto del Psion a considerar es su teclado que no es de tipo QWERTY sino de tipo ABCDE

Otro gran defecto es su pantalla, si bien es verdad que su nitidez y definición es excelente desde cualquier ángulo, no es gráfica, o sea no existe la posibilidad de crear dibujos complejos. El Psion se limita a tener un código ASCII completo, con "eñes", acentos, admiración e interrogación españoles, y una serie de caracteres definibles por el usuario programador.

Nosotros consideramos que estos defectos son mínimos, y que en conjunto el Psion es una alternativa a tener en cuenta a la hora de comprar una calculadora programable, una agenda portátil o un ordenador potente de reducido tamaño, siendo por tanto su adquisición una compra acertada.

Quien distribuye a toda España el Psion Organiser es BCN. Tel. 93-4512522, aunque está disponible en la mayoría de las tiendas de informática de cierto prestigio así como en todos los centros de El Corte Inglés.



PSION ORGANISER II adaptado a una aplicación concreta a petición del usuario.

El funcionamiento del ordenador se puede enfocar desde varios puntos de vista, la forma más adecuada podría ser haciendo una división en 6 partes, éstas serían: programas de gestión de la máquina, idioma-estadísticas de los discos (DATAPACKs y RAMPACKs) -reloj-cronómetros-alarma-temporizador, agenda, gestión de ficheros, calculadora y lenguaje de programación OPL.

```
SINPUT$:(VALIDA$,L%,X%,Y%,DEFECTO$,MODO%)
LOCAL M$(20),LO%,CURPOS%,FLAG%
LOCAL I$(20)
LO%=LEN(DEFECTO$)
CURPOS%=LO%
AT Y%,X%
PRINT DEFECTO$+REPT$(" ",L%-LEN(DEFECTO$))
M$=DEFECTO$
FLAG%=0
DO
 IF MID$(VALIDA$,CURPOS%+1,1)="9"
  KSTAT 3
 ELSEIF MID$(VALIDA$,CURPOS%+1,1)="X"
  KSTAT 1
 ENDIF
 AT Y%+CURPOS%,X%
 CURSOR ON
 CR%=GET
 CURSOR OFF
 IF CR%=8
  IF CURPOS%>0
   IF CURPOS%=LO%
    M$=MID$(M$,1,LO%-1)
    LO%=LO%-1
    CURPOS%=CURPOS%-1
    AT Y%+LO%,X%
    PRINT " "
   ELSE
    M$=MID$(M$,1,CURPOS%-1)+MID$(M$,CURPOS%+1,LO%-CURPOS%)
    LO%=LO%-1
    CURPOS%=CURPOS%-1
    AT Y%+CURPOS%,X%
    IF MODO%=3
     PRINT REPT$("#",LO%-CURPOS%);" "
    ELSE
     PRINT MID$(M$,CURPOS%+1,LO%-CURPOS%);" "
    ENDIF
   ENDIF
  ENDIF
 ELSEIF CR%=7
  IF CURPOS%<LO%
   AT Y%+CURPOS%,X%
   LO%=LO%-1
   IF MODO%=3
    PRINT REPT$("#",LO%-CURPOS%);" "
   ELSE
    PRINT MID$(M$,CURPOS%+2,LO%-CURPOS%);" ";
   ENDIF
   M$=MID$(M$,1,CURPOS%)+MID$(M$,CURPOS%+2,LO%-CURPOS%)
  ENDIF
 ELSEIF CR%=1
  M$=""
  LO%=0
  CURPOS%=0
  AT Y%,X%
  PRINT REPT$(" ",L%)
 ELSEIF CR%=3
  IF MODO%=1
   FLAG%=1
  ENDIF
 ELSEIF CR%=5
  IF CURPOS%>0
   CURPOS%=CURPOS%-1
  ENDIF
 ELSEIF CR%=6
  IF CURPOS%<LO%
   CURPOS%=CURPOS%+1
  ENDIF
 ELSEIF CR%=4
  IF MODO%=1
   FLAG%=1
  ENDIF
 ELSEIF CR%=13
  FLAG%=1
 ELSEIF CR%>=32 AND LO%<L%
  IF CURPOS%=LO%
   AT Y%+LO%,X%
   LO%=LO%+1
   CURPOS%=CURPOS%+1
   IF MODO%<>3
    PRINT CHR$(CR%);
   ELSE
    PRINT "#";
   ENDIF
   M$=MID$(M$,1,LO%-1)+CHR$(CR%)
  ELSE
   AT Y%+CURPOS%+1,X%
   IF MODO%=3
    PRINT REPT$("#",LO%-CURPOS%)
   ELSE
    PRINT MID$(M$,CURPOS%+1,LO%-CURPOS%)
   ENDIF
   AT Y%+CURPOS%,X%
   IF MODO%=3
    PRINT "#";
   ELSE
    PRINT CHR$(CR%);
   ENDIF
   LO%=LO%+1
   CURPOS%=CURPOS%+1
   I$=M$
   M$=MID$(I$,1,CURPOS%-1)+CHR$(CR%)+MID$(M$,CURPOS%,LO%-
CURPOS%)
  ENDIF
 ENDIF
UNTIL FLAG%=1
KSTAT 1
RETURN M$
```

Ejemplo de una función en OPL, esta función devuelve una cadena alfanumérica obtenida desde el teclado. Es similar a un IMPUT normal, sólo que permite asociarla a una cadena de validación (similar a los RSC de Atari), unas coordenadas, una longitud máxima y modos de ocultación.

```
XBOOT:
LOCAL A%(2)
A%(1)=$3F17
A%(2)=$3900
USR(ADDR(A%()),0)
```

Llamada directa a la ROM del sistema operativo que inicializa todos los dispositivos que estén en ese momento conectados. Normalmente los dispositivos sólo se inicializan cuando se está en el menú principal y no a mitad de programa.



entiende de alta tecnología

MIK
MEMORIA PRINCIPAL
TECLA DE FUNCIÓN
FLOPPY'S
INTERFAZ DE USUARIO
DISPOSITIVO DE ENTRADA
APPLICATION SOFTWARE (género suave de aplicación)
FLUJO DE PROGRAMA
HERRAMIENTA DE DEPURACIÓN
DISPOSITIVO DE CONTROL
DISPOSITIVO DE SALIDA
UNIDAD CENTRAL DE PROCESO
HARDWARE PERIFÉRICO
MIK y MOUSE
ERROR E/S
ATARI user
DATA COMPLEMENT.
COPIA DE SEGURIDAD

(96) 323 32 12
SOFTWARE DE DOMINIO PUBLICO APARTADO DE CORREOS 10138 46025 VALENCIA

Contactos...

Esta sección está destinada a servir de panel de anuncios entre usuarios de ATARI. Serán bienvenidos todos aquellos anuncios de compra-venta de equipos, clubs de usuarios, etc. Vuestros contactos debéis enviarlos a CBC Press, S.A. ATARI USER - CONTACTOS. Ecija, 52. Los Altos del Burgo. 28230 Las Rozas. MADRID.
Os recordamos que vender copias de software comercial es ilegal y puede ser sancionado según la legislación sobre la Propiedad Intelectual.

Tengo un 1040 STFM y me gustaría intercambiar todo tipo de información, utilidades, trucos y traducciones con otros usuarios de toda España. Respuesta asegurada. Escribe a: Ricard Forcada. Cuba, 4 1º 1ª. Vilanova i la Geltru. 08800 BARCELONA.

Vendo ATARI 520 STFM con disquetera externa, por sólo 50.000.- Ptas. Regalo juegos como: F29-Retaliator, Oro de los Aztecas, After the war y los incluidos con el ordenador. Ordenador en perfectas condiciones. Todos los manuales. Escribir a: Eliecer J. Revert Algarra. Virgen de la Salud, 12 B.A. Elda - ALICANTE. O bien llamar al teléfono 96-5391190.

Vendo Portfolio 512 16 bit PC. PC card drive, Interface card, interface cable, 3 1/2", 5 1/4" floppy disks con device driver y memory card (128), transformador format software. Todo por 80.000.- Ptas. El equipo está nuevo y sin usar. Interesados dirigirse a: Begoña Zua Zua. San Fermín, 5 - 4º drcha. 48610 Urdúliz VIZCAYA. Tel.: 94-6764454 de 3 a 4 de la tarde y después de las 10 de la noche, o al 94-4639409 en horas de comercio.

Vendo ordenador ATARI ST1040 por no usarlo, con dos posibilidades de precio: junto con 20 discos 75.000.- Ptas. Con más, 90.000.- Ptas. Todos los programas son interesantes. Escribir a: Jaime Orquín. La Palma, 29 - 2º. 03500 Benidorm - ALICANTE.

Vendo ATARI 1040 STE con 2 disqueteras , monitor monocromo SM 124, cable conexión TV color con conexión euroconector y el normal. Todo en perfecto estado, interesados dirigirse a Paco, Poeta Mas y Ros, 130 - 5ª. VALENCIA 46022. Preferible teléfono 96-371 91 86. De 21:00 a 22:00 horas.

Me gustaría contactar con usuarios del ATARI 1040 STE para intercambio de información, preferentemente música y gráficos. Mario Pou. Balmes, 67. 17600 Figueres. GIRONA.

Club Midiam PD Software busca nuevos miembros. Somos especialistas en dominio público. También buscamos «jugones» para intercambio de trucos, claves, cargadores, etc. Midiam PD Software. Antonio D. González. Conde de Peñalver, 92 - 6ºC. 28006 MADRID.

Intercambio información sobre ATARI ST. Angel B. García Magaz. Postas, 1 - 3º. 24700 Astorga - LEON. Tel.: 987-615435.
Compro Discos Duros de 10 ó 20 Megabytes, de segunda mano. Llama al 968-293985 de 4 a 6.30 de la tarde y pregunta por José Miguel.

¡Hola! Soy un usuario de ATARI 520 STFM y me gustaría contactar con otros usuarios de ST. Respuesta asegurada. Escribe a: J. M. Illán Bastida. Avda. Ronda Norte, 23 - 4º C1. 30009 MURCIA.

Vendo impresora ATARI SMM804. Compatible EPSON. En perfectas condiciones. Con alimentador, cable interface paralelo Centronics, cartucho de tinta nuevo y manual completo en castellano. Conectable a ST o cualquier ordenador con salida Centronics. Sólo 39.000 pesetas. Negociables. Escribid o llamad a José Miguel Illán Bastida. Avda. Ronda Norte, 23 - 4º C1. 30009 MURCIA. Tel.: 968-293985.

Vendo ordenador ATARI 520 STFM, embalaje original, ratón, programas originales)El Estudiante, Neochrome ...), todos los manuales necesarios completos, etc. Sólo 50.000 pesetas. Contacta con José Miguel de 4 a 6.30 de la tarde en el teléfono 968-293985.

Vendo estas últimas novedades, todas ellas originales: «RAZAS DE NOCHE, MIDNICHT RESISTANCE, REGRESO AL FUTURO II, SLY SPY. TORTUGAS NINJA, SPACE HARRIER II y DR DOOM REVENGE», por sólo 5.000.- Ptas. Interesados llamar a Diego al teléfono 924-521404.

Necesito Digitalizadores de Sonido y Vídeo, económicos con instrucciones de manejo. Enviar ofertas a Benjamín, Aptdo. 1490, LA CORUÑA. Posibilidad de compra o intercambio por programas, tanto juegos como utilidades.

Vendo monitor color Atari SC1224, con controles de brillo, volumen, contraste, amplitud, sincronismo y posición. Incluye conexión al ST y cable de alimentación, regalo manual en castellano y los magníficos juegos Xenon 2, Golden Axe y Desafío total. Todo por 30.000 pesetas. Interesados escribir a: Marcos García Valiente. Sierra Vieja, 55- 1ºB. 28031 MADRID. O llamar en horas de comida al 91 - 7860077.

Me gustaría intercambiar información sobre el STE, máxima seriedad. Interesados escribir a: Marcos García Valiente, Sierra Vieja, 55 - 1ºB. 28031 MADRID.

A Pablo y Fernando de Sestao y Ortuella, ya que criticáis a otros, ¿por qué no os aplicáis vosotros el cuento y me devolvéis mis diez discos? Más información: Miguel Suez Lago. Carlos V, 13. 03002 ALICANTE.

Contactaría con gente interesada en 68000 y programación de DEMOS. Red Devil. Aptdo. 566. 03080 ALICANTE Tel. 96 - 5218701.

Vendo ordenador ATARI MEGA ST 2, Disco Duro 30 Megas, Monitor Atari SM124, Lenguaje BASIC, programa dibujo Campus CAD (con conector para funcionar), manuales de todo incluídos. Teléfono 93 - 2582692 A. Banzo. BARCELONA.

Me gustaría contactar con usuarios del ATARI 1040 STE que lo hagan servir principalmente para música, para intercambiar ideas, experiencias, etc. Escribir a Mario Pou. Balmes, 67. 17600 Figueres. GIRONA.

ATARI 520 STE completamente nuevo, vendo. Sonido stereo y 4096 colores. Incluyo muchos juegos y utilidades, libros, un joystick y una impresora. Todo por 60.000 pesetas. Andoni Martín Arce. Rafaela Ibarra, 33 - 2°C. 48014 BILBAO. Teléfono 94 - 4477824.

Se Vende: ATARI 1040 ST, monitor SM124, PC Speed MS DOS, emulador Macintosh e impresora Epson LX-800 con introductor de hojas, programas para entorno Atari (autoedición, música, dibujo) o para PC (contabilidad, base de datos). Todo desde 110.000 pesetas. A partir de las 20 h. Tel.: 972 - 830577 Jordi - GERONA.

Me gustaría intercambiar todo tipo de información, novedades, trucos, etc., para el ATARI 1040 STE o FM. Sobre todo, programas de música y utilidades. Los interesados escribir a: Alberto M. García Lara. Pino, 27 (comercio). Andujar - JAEN. Teléfono 953 - 506697.

Intercambio información para ATARI 520 y 1040. Escribir o llamar a: Iván Fernández Azofra. Portuene, 15 - 2° izq. San Sebastián. 20008 GUIPUZCOA. Tel.: 943 - 213350.

HOJA DE SUSCRIPCION

Para recibir ATARI USER directamente sin importar el lugar o la ciudad del mundo en que resida. Suscríbase durante un año (11 números), para ello ha de rellenar este formulario y enviarlo por correo junto a un Talón o una fotocopia del resguardo de Giro Postal a C.B.C. Press.
El valor del Talón o Giro dependerá del lugar en que usted resida.

Nombre:..
Apellidos:...
Dirección:...
...
Código Postal:........................ Ciudad:............................ País: ....
Edad: .... Profesión:................................. Aficiones:..
Marca y tipo de ordenador que posee:...

Forma de pago: ☐ Talón ☐ Giro Postal
Tipo de suscripción: ☐ Nueva suscripción ☐ Renovación de la suscripción

ESPAÑA correo normal: 3.375.- Ptas.
EUROPA correo normal: 4.760.- PTAS.
AMERICA (Norte y Sur) correo normal: 5.970.- Ptas.

La dirección a la cual deben enviar su suscripción es la siguiente:

C.B.C. PRESS, S.A.
Ecija, 52
Los Altos del Burgo
28230 Las Rozas
MADRID - ESPAÑA
Para cualquier consulta llame al teléfono 91- 639 49 20 ó Fax: 91-639 51 34

Cartas

Hola amigos:
En primer lugar deseo animarles a seguir con la publicación de la revista, ya que es de gran utilidad para todos los que bien como trabajo o hobby utilizamos los magníficos ATARI.
Quisiera que me respondieran a dos cuestiones:
Primero, la ampliación de memoria a 1Mb de un 520, hace que funcione como un 1040, más concretamente se aprovecharían las mejoras de juegos como el Finest Hour, Secreto de la Isla de los Monos, etc.
Segundo, he repasado y repasado un cargador, en basic, de vidas infinitas para Nebulus y siempre me da el mismo error:
Ilegel Funcion cael at line 20
La subsodicha línea es:
READ DATO %: POKE _ W A %, DATO %
¿Podrían ayudarme? Muchas gracias.
Txomin Rekalde. Leioa - VIZCAYA.
P.D.: He comprado por primera vez la revista en el nº 24 y ya he encontrado mejoras en los dos siguientes. ¡Estupendo!

Hola Txomin, bienvenido.
Es cierto, tal y como nos cuentas que el 520 ST ampliado a 1 Mega se comporta exactamente igual a un 1040 y por tanto todos aquellos programas que estén preparados para gestionar toda la memoria harán buen uso de ésta.
Siempre podemos dividir los programas en dos grupos, los profesionales y los lúdicos, pareciendo lógico que los pertenecientes al primer grupo aprovechen toda la memoria del equipo, es entendible que en un procesador de textos como el WORDPLUS que acepte documentos mayores cuanto mayor sea la memoria del equipo.
En el grupo de los programas lúdicos, no es tan obvio que aprovechen la memoria, pero en algunos casos es así como ocurre con programas como Secreto de los Monos, Maniac Mansion, Defender of the Crown, y otros muchos. Este aprovechamiento se basa en la idea de que el juego tiene varias pantallas y cada vez que cambiamos de pantalla o fase, se carga la nueva desde el disco de modo inteligente, si tenemos memoria suficiente no borramos la pantalla antigua al cargar la nueva, y así sólo se leerá una vez cada pantalla o fase desde el disco, incluso en algunos programas se realiza una única carga inicial no utilizándose posteriormente el disco.
En otros programas, el disponer de 1Mega implica mejores sonidos y gráficos, sobre todo al principio en la presentación, como es el caso del ya mítico programa STARGLIDER II. Este grupo de programas normalmente están preparados para funcionar en un 520 STFM de simple cara, guardando en la segunda cara digitalizaciones y gráficos que sirven de presentación y sólo son cargados si el ordenador dispone al menos de 1 Megabyte y una disquetera A de doble cara.
Como es lógico, la producción de programas de juegos para el ATARI es tan grande, que siempre existen grupos de programas que no tienen en cuenta el equipo sobre el cual están funcionando, como son Operation Stealht, XENON 2, etc., suponiendo que siempre están en un 520 y por lo tanto están continuamente cargando datos del disco.
La otra pregunta que nos haces, es referente a un cargador en BASIC para el programa NEBULUS que no hemos publicado nosotros, y que por lo tanto desconocemos. No obstante, suponiendo que el listado que has tecleado sea correcto y funcione, puedes estar en uno o varios de los siguientes problemas:
- Que el cargador esté escrito para el programa Nebulus original, no funcionando correctamente si dispones de un Nebulus con origen en el Power Pack o pirata, o bien que suceda lo contrario. Si tuvieras ese problema, el cargador podría funcionar correctamente pero no tener efecto o dar un error al ser utilizado sobre una línea correcta. (Un error al dar error).
- El segundo problema, muy probable, es que el cargador esté escrito para funcionar sobre un equipo de 1 Megabyte a pesar de que el programa tan sólo requiera de 512 Kbytes. Este problema es posible, ya que un error del tipo ilegal significa que se está accediendo a una memoria que no existe con el POKE_W. Para saber si estás en este error debes mirar que A% no contenga un valor superior a 512 x 1024.
- El tercer error, poco probable, es que el intérprete o compilador de BASIC que utilizas llama Ilegal error a cualquier tipo de error que implique bombas, y un error clásico es el BUS error, que se obtiene siempre que se intenta hacer un POKE_W a una posición de memoria par. Para saber si este error existe debes mirar si la variable A% contiene valor impar justo antes de hacer un POKE.
- El cuarto error, y que seguramente tienes, es que en tu BASIC no exista la función POKE_W por tener esta otro nombre como por ejemplo POKEW. Para ello deberás buscar en tu manual de BASIC alguna función que sea POKE con alguna W alrededor.
El resto de la línea que nos has enviado parece correcta ya que de no ser así hubieras obtenido un error en el READ tipo Out of Data. Para otra vez que nos escribas, nos sería de gran ayuda que nos enviaras también el listado completo del cargador así como la instrucción exacta en la que se produce el error. Para saberla debes de dividir la línea en que se produce el error en varias líneas de una sola instrucción (ej.: 20 READ DATO% y 21 POKEW A%,DATO%). También sería interesante que nos hubieses informado sobre qué lenguaje BASIC has intentado ejecutar el cargador.
Esperamos que, en caso de que con nuestra ayuda no puedas resolver tu problema, nos vuelvas a escribir indicándonos más precisamente cómo llegas hasta él. Gracias por escribirnos y hasta la próxima.

Estimados amigos de ATARI USER.
En su magnífica publicación, ustedes comentaban en el nº 26, página 42, que existía un «poke» para los STFM, capaz de hacer trabajar a esta serie a 10 MHz, en vez de a 8 MHz., dado que es un registro «indocumentado», no lo publicó, pero, me imagino que esa razón no le impedirá acceder a mi petición. Le envío, junto a esta carta, un sobre con mi dirección y ya franqueado. Me gustaría que pusiera en él dicho «POKE». Un favor tan grande no merece un agradecimiento menor, y espero que comprenda que para un programador sin apenas recursos económicos, noticias como esta provocan taquicardias incontroladas de felicidad y un afán, yo diría casi «desesperación», por conseguir esta

Cartas

información. Me despido, agradeciendo de antemano su colaboración. Un gran saludo.
Miguel Angel Ramos. Gral. Franco, 5. 38006 S. C. de Tenerife. Tlfno. 922-276225.

Hola Miguel Angel. Conseguir que el ATARI funcione a 10 MHz suena muy bien, y es algo posible y barato y del interés de todos.
Las primeras noticias que obtuvimos de esta posibilidad fue a través de la revista alemana ST Computer de Abril de 1.988. El registro indocumentado en cuestión es el FAST-Register $FFFA30 que debe ser puesto a $0001.
Hacer que el ST funcione a 10 MHz acarrea problemas, ya que cualquier rutina que acceda a él debe poner el ATARI a 8 MHz sino se cuelga. Por ello para que el ordenador corra a 10 MHz no basta con realizar un simple POKE y ya está, sino que se requiere de un pequeño programa que intercepte las rutinas de la ROM para enlentecer y acelerar el ordenador cuando sea necesario.
No obstante, para ti y los demás lectores de ATARI USER hemos solicitado permiso a la revista alemana para publicar el listado de un accesorio que permite cambiar la velocidad del reloj, tan pronto como tengamos el permiso publicaremos un artículo dedicado a este registro indocumentado del MFP 68901 para que todos los usuarios del ST puedan beneficiarse de una mayor capacidad de cálculo.
Por ello te pedimos que tengas un poco de paciencia, que ya llegarán los 10 MHz (aunque sólo sea con 2 años de retraso) y que nos disculpes por no contestarte personalmente a una cuestión de interés común.

Estimados amigos de ATARI USER: Soy un asiduo lector de su revista. Poseo un ATARI 520 STFM Serie Oro.
Leyendo el número 25 de su revista encontré el artículo «Ampliando un 520 STFM» lo leí atentamente y haciéndoles caso abrí el ordenador para comprobar lo anteriormente leído, mi sorpresa fue que no hallé espacio dónde colocar los 16 chips LH21256-12, únicamente encontré sitio para 4 ó 5 chips y desearía saber cuáles son y si se necesitan resistencias o condensadores para que funcione.
Esperando recibir esta información, al igual que otros usuarios de este modelo de ordenador, se despide atentamente.
Rafael Gandía Carrasco. VALENCIA.

Tal y como hemos respondido a otros lectores en números anteriores, el 520 STFM serie oro se diferencia de los demás por tener sistema operativo 1.4 en dos ROMs en vez del sistema operativo 1.09 en seis ROMs y por disponer de 4 chips vacíos que deben soldarse sobre placa (no tienen agujeros) en vez de 16 chips vacíos que deben soldarse a través de la placa (el circuito impreso dispone de agujeros).
Por lo demás, el resto de los pequeños materiales es exactamente el mismo: 3 resistencias y 4 condensadores en vez de 16. OJO, algunos 520 llevan de fábrica las resistencias y los condensadores. Por otro lado siempre hay que comprar CHIPS idénticos o compatibles con lo que ya lleva el ordenador de fábrica.
Esperamos haber pulido más el polémico tema de las ampliaciones de memoria para el 520 STFM y 520 STFM serie oro.

¡Hola amigos! Ante todo mis cordiales felicitaciones por su fantástica revista.
Me gustaría que me aconsejaran sobre libros, programas, etc., para iniciarme a programar en Código Máquina. Soy un usuario de un 1040 STE, y quiero sacarle el máximo provecho a mi ordenador. A la espera de su contestación, un saludo de Arti. BARCELONA.

Estimado amigo, nos alegramos de que seas uno de nuestros muchos lectores que tienen entre sus metas domésticas el ATARI, algo que no siempre es fácil.
Para iniciarse a programar en Código Máquina has de seguir unos pasos determinados, en primer lugar debes adquirir un programa ensamblador, nosotros recomendamos el DEVPAC ST de la casa HISOFT que no es más que un macroensamblador capaz de ejecutar los programas paso a paso. Este programa lo puedes solicitar directamente a HISOFT en Inglaterra o bien importarlo a través de cualquier distribuidor español, como los que se anuncian en estas páginas.
El DEVPAC quieras o no quieras se vende en la actualidad en inglés, con manual en inglés, lo cual es un problema sino te defiendes en esta lengua. Tenemos noticias, que pronto algunos productos de la casa HISOFT, entre los que están el HISOFT BASIC y el DEVPAC serán traducidos al castellano.
En segundo lugar debes hacerte con algún libro de programación del 68000 simultáneamente con algún libro del sistema operativo del ST. Los libros que hablan del 68000 en castellano son varios, entre los que destacan el «68000, 68010/68020 Arquitectura y programación en ensamblador» de Stan Kelly-Bootle y Bob Fowler editado por ANAYA MULTIMEDIA y «Manual del Microprocesador 68000, 68008, 68010 y 68020» de Willian Cramer y Gerry Kane editado por OSBORNE/McGraw-Hill. El primero consta de unas 380 páginas, es fácil de encontrar y cuesta alrededor de 3.800 pesetas. El segundo libro cuenta con tan sólo 150 páginas y es más difícil de encontrar, su precio es de unas 3.400 pesetas.
El primer libro es mucho más básico que el segundo siguiendo un estilo similar a cualquier libro de texto, a pesar de tener una estructura correcta es muy lento de leer (dedica muchas páginas a cada nueva idea). Es un libro ideal para aquellos lectores que estén acostumbrados a leer libros pesados y estudiar. En cambio el segundo libro, es un libro rápido de leer, escrito en un estilo americano muy intuitivo, en el cual curiosamente se dedica su mayor parte a estudiar el 68000 visto desde el Hardware. A nuestro juicio el segundo libro es mejor y más completo, aunque sólo lo aconsejamos a aquellos que ya tengan unas nociones iniciales sobre código máquina, mientras el de ANAYA está al alcance de cualquiera.
Si desearas profundizar más, te recomendaríamos que solicitases «Programer´s Reference Manual 68000, 68008, 68010, 68012, 68020, 68030, 68040, 68851, 68881, 68882 y CPU 32 de la propia casa MOTOROLA, este libro al no existir en España, deberás solicitarlo a Motorola Ltd., European Literature Center, 88 Tanners Drive,

Cartas

Blakelands, Milton Keynes, MK14 5BP, England. En este último libro se describe lo necesario para sacar lo máximo, en cuanto a software se refiere, a cualquier procesador de la serie 68000 incluyendo la combinación 68030+68882 que posee el ATARI TT030.

Volviendo a los libros que hablan sobre el ATARI ST, te aconsejamos que compres la versión inglesa (la original está en alemán) de un libro llamado «ATARI ST Internals» de K. Gerits, L. English, R. Bruckmann editado por Abacus Software (ISBN 0-916439-46-1) y lo puedes pedir a DATA Becker GmbH, Merowingerstr. 30, 4000 Düsseldorf, Alemania o bien a ABACUS Software Inc, P.O. Box 7219, Grand Rapids, MI 49510, Estados Unidos. En este libro se describe con cierta seriedad todas las llamadas del BIOS, XBIOS y GEMDOS (dejando el GEM de lado). Este libro, cada vez que aparece una ROM nueva se reescriben las partes a fin de adaptarse al último sistema operativo.

No obstante existe una traducción de una de las primeras versiones del ST INTERNALS, llamada ST Interno publicada por Ferre Moret, S.A. C/ Córcega, 299. 08008 BARCELONA. No recomendamos esa versión castellana, por su obsolencia, a no ser que por motivo de idioma no quede más remedio.

Si quieres documentación bruta, puedes solicitar el paquete de desarrollo ATARI, que con sus más de 2.500 páginas y 10 discos contienen toda la información. Esta documentación está escrita en inglés y no son un libro ni un manual, sino una sucesión de documentos escritos aparentemente por los propios programadores de ATARI para las casas de software.

Sobre este paquete de desarrollo, que puedes comprar en España a través de Ordenadores ATARI, S.A., puedes averiguarlo absolutamente todo. Por cierto que en los 10 discos que lo acompañan hay un compilador de C, un ensamblador y muchos ejemplos, aunque todo de pésima calidad a pesar de que sea el utilizado por ATARI inicialmente para programar la ROM, lo único que vale la pena es la documentación. (Asembler malo, C malo, Editor de RSC malo, ...).

Una vez que poseas estas herramientas, no te será difícil aprender Código Máquina, incluso tú mismo podrás ir encontrando libros que hablen del GEM, Unidades de Disco, Gráficos en 3D, etc. Aún así, en este número iniciamos en ATARI USER un curso sobre Código Máquina, que partiendo de cero, permitirá que cualquier lector pueda programar en este lenguaje antes de que termine el verano.

Esperamos haberte abierto el sendero que puede llevar a un usuario a dominar la máquina.

Hola soy un lector de esta fabulosa revista como es ATARI USER. Tengo un problema con mi impresora, esta es la Sprinter 180: Schneider Rdf. AG.

Mi ordenador es un ATARI 520 STFM. El problema radica en que no sé como sacar gráficos, pero si sé que se puede hacer con estos instrumentos. Yo creo que la solución esta en la configuración y en un código que hay que poner para que la impresora entienda el código del ordenador.

También me gustaría saber si en las Palmas de Gran Canaria hay tiendas especializadas en ordenadores y más concretamente en juegos Atari. Si me dan la dirección les quedaría muy agradecido.

Muchas gracias por haber contestado a mi carta y saludos.

Estimado lector, sentimos tener que decirte que desconocemos la impresora Sprinter 180, y que por lo tanto no creemos que nuestra ayuda te sea eficaz, por ello te sugerimos que nos vuelvas a escribir adjuntando fotocopias del manual de programación de tu impresora.

No obstante podemos suponer que tu impresora es una semicompatible EPSON, que funciona correctamente al imprimir textos ASCII y que falla cuando pulsas ALTERNATE - HELP. En algunos casos, se puede conseguir que funcione la impresora correctamente, instalando con ayuda del panel de control la impresora. Intenta instalarla a 1200 puntos o 960 puntos, a fin que la trate como una EPSON y no una SMM 804.

Esperamos con impaciencia una nueva carta tuya en la que nos remitas el manual de instrucciones a fin de que podamos ayudarte. Hasta la próxima.

Ah! La dirección que nos pides de algún centro ATARI en Las Palmas es la siguiente:

INCO INSULAR COMPUTER. José M. Motas, 23. 35016 Las Palmas de Gran Canaria. Tel.-Fax: 928-322445.

Saludos, colegas de Atari User. Os escribo para plantearos algunas cuestiones.

Soy un nuevo usuario de un ATARI 1040 STE y estoy super-interesado en la programación ensamblador del 68000, especialmente enfocada a tratamientos gráficos y sonoro (sprites, scrolles, colisiones, etc.).

1.- ¿En qué posiciones de memoria se encuentra la pantalla gráfica del Atari y cual es su arquitectura?

2.- ¿Cómo se pueden manejar desde el lenguaje ensamblador las nuevas facilidades del STE (Shifter, Blitter) y cómo funcionan exactamente?

3.- ¿Una vez escrito un programa en Gens y Ensamblador, cómo puedo ejecutarlo directamente desde el GEM?

4.- ¿Cómo funciona el chip de sonido del STE (si es mucho pedir, al menos el normal, el del FM, que tengo entendido se incluye en el STE) y cómo se maneja desde Código Máquina?

5.- ¿Dónde podría encontrar un «buen» libro en que se detalle la arquitectura interna (tanto a nivel de Soft como Hard) del Atari STE (no importa que sea en inglés)?

6.- Desearía que pusierais más énfasis en la revista acerca de los temas relacionados con programación del Atari a bajo nivel, que creo es donde se puede sacar más partido a esta estupenda máquina (interrupciones, rutinas ROM, variables internas, etc.).

Muchísimas gracias y felicidades por la revista.

Alvaro Durán Martínez
F. Arraijanal Ctra. Coin s/n.
29120 Alhaurin el Grande - MALAGA

P.D. Desearía contactar con usuarios interesados en todo este tema.

Vemos que eres uno de nuestros muchos lectores interesados en domesticar su ST al más bajo nivel, obligándole a hacer exactamente lo que tú quieras. A continuación contestamos brevemente a tus preguntas:

1.- El ATARI ST a diferencia de otros ordenadores más antiguos como el

Cartas

Spectrum no tiene la pantalla en una dirección de memoria fija, sino que la puedes poner en la dirección de memoria que más te convenga, siempre y cuando ésta sea divisible por 256 (en un ST FM), por 8 (en un TT) y por 2 (en un STE).

El ATARI no sólo tiene una pantalla en una dirección de memoria definible por el usuario, sino que además se puede considerar que en todo momento existen dos pantallas llamadas lógica y física. La pantalla lógica es en la que se escriben y dibujan las cosas, mientras que la pantalla física es la que muestra el monitor. Normalmente, la pantalla lógica y física están ubicadas en la misma posición a fin de que se muestre en el monitor lo que se va dibujando sobre ella, pero en muchos programas, generalmente juegos, demos y animación se suele tener la pantalla física y lógica en diferentes posiciones a fin de mostrar una pantalla mientras se dibuja la otra para finalmente invertirlas, esto es cambiar la lógica por física. De este modo podemos realizar animaciones (Sprites y Scrolles) sin que exista ningún tipo de parpadeo u efecto secundario.

Como es lógico, cuando se arranca el ordenador el sistema reserva un espacio de memoria de 32 Kbytes para albergar la pantalla (150 en el TT). Podemos averiguar su posición en memoria realizando una llamada al XBIOS de la ROM, tal y como sigue:

```
; nos devuelve la posición de la pantalla física.
physbase:
move.w   #2, -(A7)  ; Función 2 del XBIOS.
trap     #14        ; Ejecuta el TRAP #14 (XBIOS).
addq.1   #2,A7      ; Pone la pila a su valor inicial.
;D0.L contiene la posición de la pantalla física.
logbase:
move.w   #3, -(A7)  ; Función 3 del XBIOS.
trap     #14        ; Ejecuta el TRAP #14 (XBIOS).
addq.1   #2,A7      ; Pone la pila a su valor inicial.
;D0.L contiene la posición de la pantalla lógica.
```

En este primer punto, también nos pides cuál es la estructura de la pantalla del ATARI, lo cual intentamos explicarte resumidamente en las siguientes líneas.

En primer lugar, tienes que tener en cuenta que cualquier ST dispone de 3 resoluciones distintas, una de 320x200 a 16 colores, otra de 640x200 a 4 colores y una última de 640x400 monocroma. (Nos olvidamos por el momento de las resoluciones existentes en el TT). Cada resolución tiene una estructura propia, aunque todas comparten algo en común. Todas tienen una estructura de 16 bits, esto es la primera palabra, igual a 2 bytes, que corresponden a los 16 primeros pixels.

El significado de la segunda palabra depende de la resolución en la que nos encontremos. Para obtener 16 colores, numerados del 0 al 15 necesitamos al menos un número de 4 bits con lo cual un modo con 16 colores obliga forzosamente a tener lo que se denominan 4 bitplanes, mientras un modo de 4 colores sólo exige 2 bitplanes y uno de 2 colores sólo exige un bitplane.

Esto quiere decir que en un modo de 16 colores la segunda, tercera y cuarta palabra, al igual que la primera corresponde a los 16 primeros pixels, mientras la quinta, sexta, séptima y octava corresponden a los siguientes 16 pixels ... Si tuviéramos un solo bitplane la primera palabra correspondería a los 16 primeros pixels, la segunda a los 16 siguientes ...

Para conocer cuántas palabras ocupa una línea, sólo tenemos que dividir su resolución por 16 y multiplicarla por el número de bitplanes que posee. En el modo 320x200 a 16 colores ocupa 320/16*4=80 palabras.

De este modo las primeras 80 palabras, corresponden a la primera línea, las segundas 80 palabras a la segunda línea, y así para todas las líneas. (En monocromo sería cada 640/16*1=40 palabras). Si tenemos 200 líneas te darás cuenta que la pantalla ocupa exactamente 32000 bytes. En un TT con resolución 320x480 a 256 colores (8 bitplanes), cada línea ocupará 320/16*8=160 palabras=320 bytes/línea y como hay 480 líneas ocupa en total 153600 bytes.

Para completar cuál es el significado de dichas palabras debes imaginarte, que una resolución con 4 bitplanes, el primer bit de la primera palabra más el primer bit de la segunda palabra más el primer bit de la tercera palabra más el primer bit de la cuarta palabra forman una cifra de 4 bits en la cual se puede representar un número del 0 al 15 que corresponda al color del primer pixel. Lo mismo es válido para los segundos y siguientes bits.

Para complicar un poco más el asunto, en el ATARI a diferencia de otros ordenadores más antiguos como el IBM PC con tarjeta CGA (léase AMSTRAD 1512), los colores pueden ser definidos a voluntad del programador, pudiendo ser el color 0 un blanco, rojo o azul según consideremos oportuno. (Sobre el detalle de las paletas podrás encontrar una explicación más intuitiva en la sección de cartas de ATARI USER nº 25 página 47). El ATARI dispone de una paleta de 512 colores (4096 en STE) de la cual sólo podemos mostrar 4, 16 ó 256 según dispongamos de 2, 4 u 8 bitplanes. En un STFM y un STE la paleta de colores está en la posición de memoria $ff8240 y consta de 16 palabras.

En cada palabra debe ir un número hexadecimal de 3 cifras, siendo la de la derecha intensidad de azul, la de en medio intensidad de verde y la de la izquierda intensidad de rojo. En un STFM cada cifra puede tomar un valor entre 0 y 7, mientras en el STE puede tomar un valor entre 0 y F, pero teniendo en cuenta que los valores superiores a 7 son similares a los mismos restándoles 7 aunque un punto más oscuro. De este modo $800 es un rojo más oscuro que $100. Por último, te informamos que el borde tiene normalmente color 0, la excepción está en los monitores monocromo, que el borde es siempre negro.

A los valores $ff8240, al igual que todos los registros de hardware, sólo se puede acceder a ellos en modo supervisor, el siguiente ejemplo pone el borde de tu ordenador de color azul marino, asignando dicho color al color 0.

```
ejemplo: ; Pone el borde de azul marino.
move.1  #0, -(A7)
move.w #$20, -(A7) ; Función S20 del GEMDOS.
trap    #1         ; Ejecuta TRAP #1 (GEMDOS).
addq.1  #6,A7      ; Pone la pila a su valor inicial.
move.1  D0, pila   ; Guarda la pila usuario.

move.w  #$00F, $ff8240  ; Define el color 0.

move.1  pila, -(A7)  ; Vuelve a modo
```

Cartas

```
usuario.
move.w #$20, -(A7) ; Función $20 del GEMDOS.
trap   #1          ; Ejecuta TRAP #1 (GEMDOS).
addq.1 #6,A7       ; Pone la pila a su valor inicial.
move.1 D0, pila    ; Guarda la pila usuario.

clr.w  -(A7)       ; Sale del programa con
trap   #1          ; una llamada al GEMDOS.

pila: dc.1 0       ; Aquí se guarda temporalmente
                   ; la pila de usuario.
```

Para terminar este punto, puedes cambiar la dirección de la pantalla con la siguiente llamada al XBIOS:

```
posición:
move.w res,-(A7)      ; Resolución deseada.
move.1 physadr,-(A7); Dirección de la pantalla física.
move.1 logadr,-(A7)   ; Dirección de la pantalla lógica.
move.w #5,-(A7)       ; Llamada a la función 5 de la XBIOS.
trap   #14            ; Ejecuta el TRAP #14.
add.1  #12,A7         ; Ajusta la pila.
```

Si sólo deseas cambiar uno de estos parámetros (resolución, dirección física y dirección lógica) debes de poner su nuevo valor en lugar de res, physadr o logadr y #-1 en los otros. (#-1 indica al sistema no alterar). (res=0 320x200, res=1 640x200 y res=2 640x400).

2.- Tu segunda pregunta es más sencilla de responder, las nuevas posibilidades de SHIFTER las puedes utilizar haciendo uso de la paleta extendida (valores superiores a 7) o bien sacar provecho de pantallas virtuales tal y como expusimos en la sección de cartas del nº 27 de ATARI USER en la página 42, donde están además expuestos todos los registros del STE.

La utilización directa del Blitter es algo más complicado de explicar y no podemos hacerlo en la sección de cartas. Lo único que podemos decirte es que todas las rutinas de la ROM que acceden a pantalla, incluyendo las del A-Line hacen uso del Blitter. Para no desanimarte procuraremos tan pronto como nos sea posible dedicar unas cuantas páginas a este chip.

3.- Cuando nos mencionas el «GENST», imaginamos que te refieres al ensamblador de HISOFT. Este en su versión completa, no la de dominio público, puede ensamblar a memoria o a disco con la opción Assemble. Si hacemos que ensamble a disco con un fichero acabado en .PRG, .TOS o .TTP en formato ejecutable podrás luego ejecutarlo desde el GEM. Si lo ensamblas en memoria sólo podrás ejecutarlo desde el GENST pulsando ALTERNATE X.

4.- El funcionamiento del CHIP de sonido DMA (el del STE) lo tienes explicado en la sección de cartas de ATARI USER nº 26. El otro, el del STFM es mucho más complejo, ya que se utiliza entre otras cosas para seleccionar unidad de disco y enviar caracteres a la impresora.

El chip de sonido del STFM, que está incluido en el STE es el YM2149, creado inicialmente para máquinas de bar aunque se utiliza en muchos ordenadores distintos como los Spectrum +2 y Commodore 64. Este chip consta de 16 registros de 8 bits, numerados del 0 al 15 de los cuales los 14 primeros son los que se utilizan para crear sonidos.

Los registros 0,1 determinan la frecuencia (en realidad el periodo) y el tono del CANAL A. El registro 0 debe contener el periodo mientras el registro 1 (sólo se usan 4 bits) debe contener el tono. Los registros 2, 3 son similares al 0, 1 pero para el CANAL B, los registros 4, 5 lo mismo para el CANAL C. Un valor bajo en el tono equivale a un tono alto.

El registro 6 (sólo se usan 5 bits) es el registro de control de ruido, un valor bajo mucho ruido, un valor alto poco ruido. El registro 7 es una máscara de 8 bits, en los cuales el bit 0 indica tono del canal A, el 1 del canal B y el 2 del canal C. Los bits 3, 4 y 5 lo mismo pero para los generadores de ruido de los distintos canales. (Si queremos que se genere tono y/o ruido debemos poner a 1 los bits adecuados).

El registro 8 es el volumen, los bits 0, 1 y 2 representan el volumen, mientras el bit 4 es una máscara que indica si debe haber envolvente o no. Si hay envolvente el volumen carecerá de significado ya que se utilizará el de la envolvente. Los registros 9 y 10 son similares al 8 sólo que para los canales B y C.

Los registros 11 y 12 contienen una palabra que indica la velocidad de la envolvente. Esta palabra está en formato ZILOG o INTEL, esto es que el byte alto está en el registro 11 y el bajo en el 12. Por último tenemos el registro 13, que es el que indica al chip el tipo de envolvente que deseamos generar pudiendo tomar valores 0, 4, 8 - 15, dándonos subidas, bajadas, sierras, ...

Para crear un sonido, tan sólo debemos colocar valores adecuados en el chip de sonido, para ello, en modo supervisor debemos de escribir en la posición de memoria $ff8800 el número de registro que deseamos alterar y a continuación su valor en $ff8802, por ejemplo si deseamos meter el valor 20 en registro 6 deberemos:

```
move.b #06,$ff8800 ; Seleccionamos registro.
move.b #20,$ff8802 ; Asignamos valor.
```

Si deseamos leer el valor de los registros, tan sólo debemos seleccionar registro y leer su valor con move.b $ff8800,D0. Te recomendamos que hagas tantas pruebas como te sea posible, sin desanimarte cada vez que cuelgues el ordenador.

5.- La única documentación existente sobre el STE en España está en inglés, se trata de una ampliación al paquete de desarrollo de ATARI, y lo puedes solicitar a Ordenadores ATARI, S.A. Apartado 195, Alcobendas, 28100 Madrid, Tel.: 91-661 56 25. En dicha ampliación se comentan todos los registros de hardware nuevos del STE además de tener un número limitado de ejemplos. Fuera de España puedes buscar un libro titulado «ST Internals» de Abacus Software, que en su última edición repasa las posibilidades del STE. Este libro, por su gran interés quizás lo puedes encontrar en algún comercio especializado en ordenadores ATARI. Por otra parte, rebuscando en librerías puedes encontrar dos buenos libros en castellano dedicados al ATARI, el «ST INTERNO», que nos es más que una traducción de la primera edición del «ST INTERNALS» y «ATARI ST Consejos y Trucos», el cual incorpora bastantes listados en ensamblador. Ambos libros son antiguos (año 1.986) y son de la editorial Ferre Moret S.A. y si se encuentran, a buenos precios como restos de serie. Nosotros adquirimos el otro día 9 ejemplares en una librería a 150 pesetas ejemplar, aunque sus precios originales son de 3.000 pesetas ejemplar.

6.- Tomamos nota de tus sugerencias, espera a los próximos números.

Esperamos haber podido ayudarte aunque sea de forma escueta, por desgracia, y en contra de nuestra voluntad ATARI

Un poco de ...

USER tan sólo cuenta con 52 páginas por lo cual no podemos publicar todos nuestros textos, dibujos y fotografías, quizás, quizás pronto dispongamos de más páginas. No obstante ATARI USER dedicará en el futuro gran parte de la revista a los lectores , que como tú, deseen sacar el máximo partido de su ordenador a través de los lenguajes de programación, convirtiéndose de este modo en un servicio post-venta de incalculable valor, que ATARI, pese a su buena voluntad, no siempre está dispuesto a ofrecer a los usuarios de lengua hispana y que a la larga saldrá beneficiada gracias a la nueva generación de programadores que puede surgir de aquí, creándose indirectamente de este modo una industria de Software que probablemente terminará con la carencia del mismo.

EL RINCON

No hace falta decir que cada uno de vosotros puede hacer uso de, al menos, una columna de la revista. Podéis hacer cualquier tipo de comentario, ¡Adelante!

Ya era hora que al Mega ST le tocara el turno de la renovación, y vaya que lo han conseguido con el nuevo Mega STE, que a parte de cambiar su estética, incorpora una serie de complementos, como son el disco duro de serie, con la norma SCSI (se lee «escasi» y es utilizado, entre otros por APPLE e IBM), hardware preparado para trabajar con redes locales (LOCALTALK), memoria caché, el nuevo bus VME y todas las lindezas de la serie STE. Una verdadera maravilla, sólo falta que los distribuidores de ATARI hagan gala de buen gusto y se copien (en el mejor sentido de la palabra) de los COMMODORE, recomprando, a buen precio, nuestros antiguos ST´s, a la compra de un nuevo MEGA o TT. Por una vez copiemos una buena idea.
Dentro de poco vamos a tener que enterarnos de las novedades de nuestro ST, por las revistas extranjeras, dado el estilo propangadístico de tiendas y distribuidores del país, y no es que defienda la publicidad en masa (en cuanto al número de páginas, claro), pero no estaría de más promocionar algún producto nuevo de vez en cuando ¿acaso no nos hace falta? ¿tal vez se venden solos? No lo creo, después que no se quejen de que tal o cual producto no alcanza las cotas de venta estimadas.
En otro orden de cosas, hay que destacar la buena labor desarrollada por la BBS de ATARI, ATARI-COMM, tanto por la presentación en pantalla (menús en castellano) como por el celo en no permitir el uso ilegal de software en la BBS. Hasta tal punto que una revista, no relacionada directamente con el mundo ATARI, publica un artículo sobre su funcionamiento. Y es que hay que ser realistas, la piratería puede hundir a cualquier ordenador, y el nuestro no es una excepción. Compremos programas originales con lo cual todos saldremos ganando, aunque como dije en mi rincón anterior, la culpa no sólo es nuestra.

José Albalat Boira.

FE DE ERRATAS

En la página 42, de ATARI USER Nº 27, en el listado de la colaboración sobre FRACTALES, de nuestro amigo RAMIRO TELLEZ SANZ se perdieron, al hacer la reproducción parte de las líneas finales del mismo. Para aquellos que hayáis probado a teclearlo debéis añadir las líneas que a continuación transcribimos. Disculpas .

```
PRINT "Fin, Pulsa una tecla"
VOID INP (2)
PROCEDURE pos
an=an+z
RETURN
PROCEDURE neg
an=an-z
RETURN
PROCEDURE recta
xx=x+h*COSQ (an)
yy=y+h*SINQ (an)
RETURN
PROCEDURE xy
x=xx
y=yy
RETURN
PROCEDURE inserción
a$=LEFT$ (a$, n-1)
+q$+RIGHT$(a$, 1-n)
n=n+LEN(q$)-1
1=LEN(a$)
RETURN
```

MADRID 28013 Plaza Callao, 1 - 1ª. (91) 521 22 54 / 521 26 82
BARCELONA 08025 Pi i Margall, 58 - 60. (93) 210 68 23 / 213 42 37
BILBAO 48001 Ledesma, 4 - 2ª. (94) 424 36 68 / 424 36 70



MUSICA

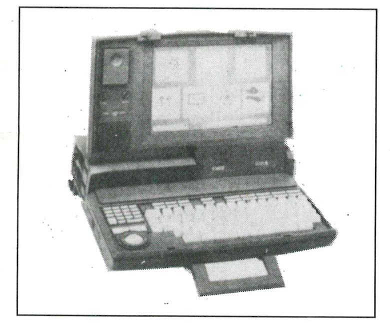
ATARI STE 1040
70.000 Ptas. + MONITOR B/N = 90.000 Ptas.

MEGA STE2 (48 Mb HD) + MONITOR B/N
+ C-LAB NOTATOR 3.0 ó CUBASE 2.0
+ IMPRESORA NEC P20 ó EPSON LQ500

317.000 Ptas.

ATARI 1040 STE
+
MONITOR B/N
+
C-LAB NOTATOR 3.0

167.000 Ptas.



LA SOLUCION PORTABLE

STACY 2/20, LCD + 2Mb + HD 20 Mb	280.000.-
STACY 4/20, LCD + 4MB + HD 40 MB	350.000.-
+ C-LAB NOTATOR 3.0 ó CUBASE 2.0	+ 72.000.-

EMULACION

ATonce Plus	SUPER CHARGER	SPECTRE GCR 3.1
PC 286 16 Mhz. Interno Indice Norton 8.0	PC 286 16 Mhz. Externo Indice Norton 14 1 Mb RAM	MAC c/ROM + Rápido + Compatible Soportado TT
46.000.-	96.000.-	72.000.-

OFERTAS

AMPLIACION 520 STE
¡¡ 8.000 Ptas. !!

LYNX
EL PRIMER SISTEMA PORTATIL DE ENTRETENIMIENTO EN COLOR
¡ DESDE 16.000 Ptas. !

AMPLIACION 520 STE
¡¡ 8.000 Ptas. !!

PORTFOLIO
EL PRIMER PC COMPATIBLE DE BOLSILLO
MS-DOS, Agenda, Diario, Hoja de Cálculo, Calculadora y Procesador de texto
¡ DESDE 44.500 Ptas. !

IMPRESORA CHORRO DE TINTA
HEWLETT - PACKARD DESKJET 500
¡¡ 80.000 Ptas. !!

MODEM 2400 bd.
¡ SOLO 20.000 Ptas. !

ATARI® *Portfolio*

UN VERDADERO ORDENADOR COMPATIBLE PC DE BOLSILLO, QUE SE PUEDE CONECTAR A UNA IMPRESORA O INTERCAMBIAR INFORMACION CON OTROS ORDENADORES, PARA PODER TRABAJAR EN CUALQUIER MOMENTO Y EN CUALQUIER LUGAR, EL AVION, LA OFICINA, EL COCHE, LA UNIVERSIDAD...

TODA LA POTENCIA DE UN ORDENADOR AHORA EN SUS MANOS.



CARACTERÍSTICAS TÉCNICAS

- Procesador: INTEL 80c88 (de bajo consumo), frecuencia de reloj: 4,91 Mhz.
- Memoria: 128 Kb. expandible a 640 Kb.
- Compatibilidad: con el sistema operativo MS-DOS (V.2.11).
- ROM: 256 Kb. con software integrado.
- Bus de expansión y conexiones: de 60 pines para interfaces RS 232 y Centronics combinados, expansión de la RAM, comunicación con otro PC, conexión para impresora.
- Medio de almacenamiento: tarjetas RAM en formato Tarjeta de crédito.
- Dimensiones: 18 × 9 × 2,5 cm.
- Peso: 450 gramos (incluyendo las pilas).

INCLUYE

- Editor de Textos.
- Hoja de Cálculo compatible con Lotus 1-2-3.
- Agenda, consistente en un Dietario, Fichero de Direcciones y Calendario para los próximos 60 años.





ORDENADORES ATARI, S. A. Apartado 195 • Alcobendas, 28100 Madrid • Telf. (91) 661 56 25

NUMERO UNO COMUNICACION